

OpenText™ Fortify Software Security Center

Software Version: 24.4.0

Database Performance and Maintenance Guide

Document Release Date: October 2024

Software Release Date: October 2024

Legal Notices

Open Text Corporation

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

Copyright Notice

Copyright 2008 - 2024 Open Text.

The only warranties for products and services of Open Text and its affiliates and licensors (“Open Text”) are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Trademark Notices

“OpenText” and other Open Text trademarks and service marks are the property of Open Text or its affiliates. All other trademarks or service marks are the property of their respective owners.

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 for OpenText™ Fortify Software Security Center CE 24.4 on October 30, 2024. 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	4
Contacting Customer Support	4
For More Information	4
About the Documentation Set	4
Fortify Product Feature Videos	4
Chapter 1: About This Guide	5
Chapter 2: Fortify Software Security Center Database Setting Recommendations	6
Microsoft SQL Server Settings	6
Chapter 3: Database Performance Issues	12
Disk I/O	12
Oracle Databases: Automatic Workload Repository for Database Tuning	13
MySQL Databases: Percona Toolkit	15
MySQL InnoDB Buffer Pool	15
Microsoft SQL Server	15
Indexing Fragmentation	17
Fortify Software Security Center Scheduler	17
Managing Authentication Tokens	19
Managing Artifacts	19
Tables for Removing Data from the Fortify Software Security Center Database	29
Maintenance Schedule	30
Appendix A: Database Queries: MS SQL (On Prem)	31

Preface

Contacting Customer Support

Visit the Support website to:

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

<https://www.microfocus.com/support>

For More Information

For more information about Fortify software products:

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

About the Documentation Set

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

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

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

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

Fortify Product Feature Videos

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

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

Chapter 1: About This Guide

This guide is intended to help users with Fortify Software Security Center implementations maintain and adjust the Fortify Software Security Center database. This information is intended to help you understand the database solution you use with Fortify Software Security Center and the situations that warrant adjustments to and maintenance of your database.

OpenText strongly recommends that you work with your DBA to ensure that all actions are performed correctly and securely. If you do not have access to a DBA, Professional Services can help you tune and maintain your Fortify Software Security Center database.

Caution! Fortify Software Security Center does not support MS SQL or Oracle in the cloud.

This document includes the following information:

- ["Fortify Software Security Center Database Setting Recommendations" on page 6](#) - The tables in this section provide guidance and recommendations on database hardware requirements based on the database type and metrics collected.
- ["Database Performance Issues" on page 12](#) provides information about:
 - Disk I/O (["Disk I/O" on page 12](#)) issues related to performance
 - Indexing issues (["Indexing Fragmentation " on page 17](#))
 - Scheduler options for data retention (["Fortify Software Security Center Scheduler" on page 17](#))
 - Managing artifacts to keep the database lean (["Managing Artifacts" on page 19](#))
 - Scheduling database maintenance (["Maintenance Schedule" on page 30](#))
- The appendix (["Database Queries: MS SQL \(On Prem\)" on page 31](#)) includes:
 - ["Database Queries: MS SQL \(On Prem\)" on page 31](#) lists MS SQL queries for your DBA team to execute. OpenText recommends that these queries be executed by your database administrator. If you are a new Fortify Software Security Center user, these queries must be run to establish a baseline. As your database grows, and as it approaches 1 TB in size, consider re-running the queries and comparing the data to the baseline data. (An output example and an description of what to look for are provided for each query.)

If you are an experienced Fortify Software Security Center user, and you are seeing performance issues, use these queries to collect the necessary data so that Customer Support can use it to provide feedback and recommendations.
 - ["Database Queries: MS SQL \(On Prem\)" on page 31](#) provides suggestions on how to query MS SQL databases.
 - ["Database Queries: MS SQL \(On Prem\)" on page 31](#) provides suggestions on how to query Oracle databases.

Chapter 2: Fortify Software Security Center Database Setting Recommendations

The following sections provide guidance and recommendations for your database, based on database type and metrics.

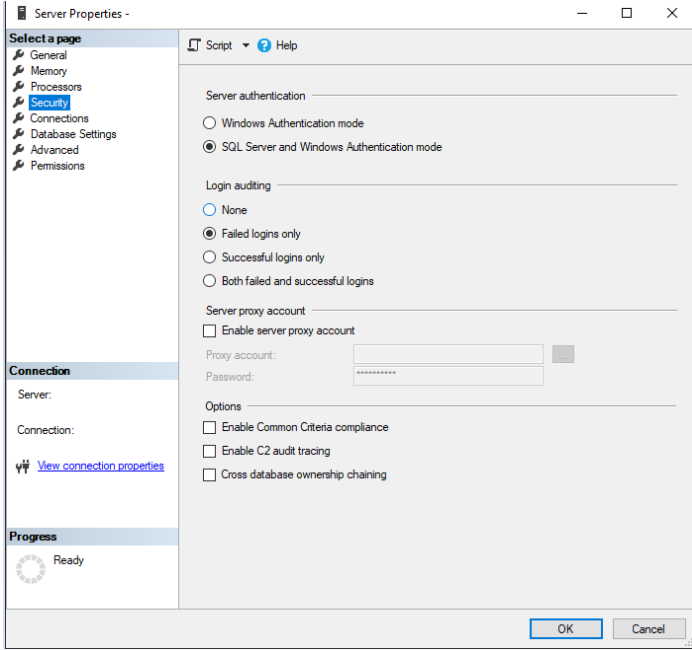
Microsoft SQL Server Settings

The Microsoft SQL "Cost Threshold for Parallelism" property determines the threshold at which SQL Server creates and runs parallel plans for queries. SQL Server creates and runs a parallel plan for a query only if the estimated cost of running a serial plan for the same query is higher than the value set for Cost Threshold for Parallelism property.

Note: The "cost" refers to the estimated cost of running the serial plan on a specific hardware configuration, and does *not* refer to a unit of time.

Server Properties Security Settings

OpenText recommends the following Microsoft SQL Server Properties Security settings.



Server authentication section

- Option 1: Windows Authentication mode: OpenText does not recommend “windows only” mode as it limits the ability for SSC to operate in a Linux environment.

- Option 2: SQL Server and Windows Authentication mode: OpenText supports and recommends this “mixed” mode for server authentication.

Login auditing section

- Fortify only uses and validates the “Failed logins only” option. However, you may choose to audit logins. Fortify does not dictate this setting. None of the options will have an effect on your Fortify implementation.

Server proxy account section

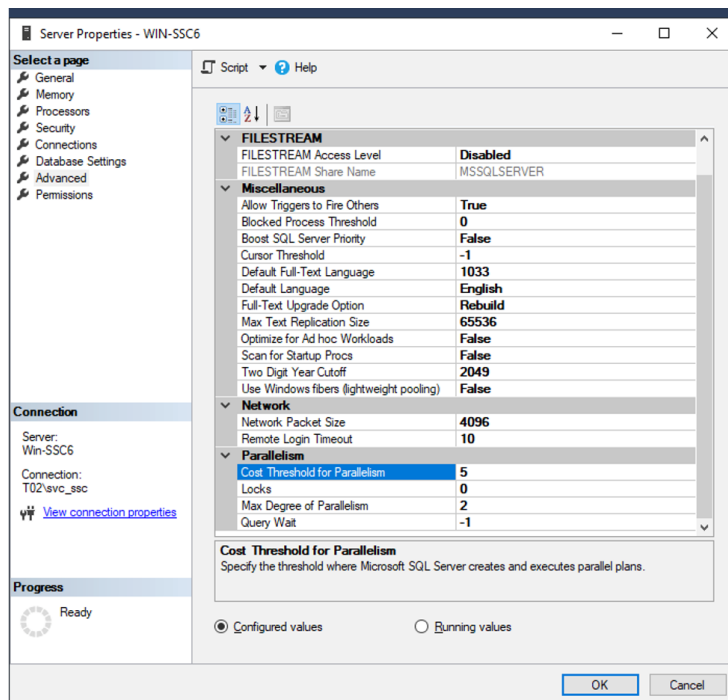
Fortify does not Enable server proxy accounts internally. Therefore, Fortify does not provide guidance on this setting.

Options section

Fortify does not select or test any of the options in this section. Therefore, Fortify does not provide guidance on these settings.

Server Properties Advanced Settings

OpenText recommends the following Microsoft SQL Server Properties Advanced settings.



The default value for Cost Threshold for Parallelism is 5, which means that the optimizer switches to a parallel plan if the cost of a query plan is more than 5. You can set this property to any value from 0 through 32767. Microsoft recommends that the property be set only by an experienced database administrator or a certified SQL Server professional.

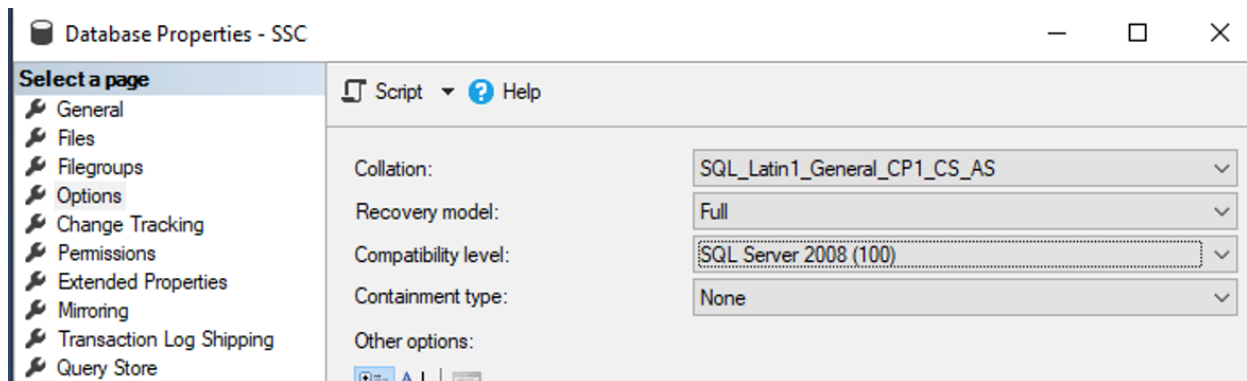
If the Max Degree of Parallelism property is set to 1, SQL ignores the value set for Cost Threshold of Parallelism. OpenText recommends that you change this value to 50. Parallelism occurs even if you

increase the Max Degree of Parallelism value. The aim is to minimize unwanted parallelism. SQL waits for the data to be returned from the queries that have gone parallel.

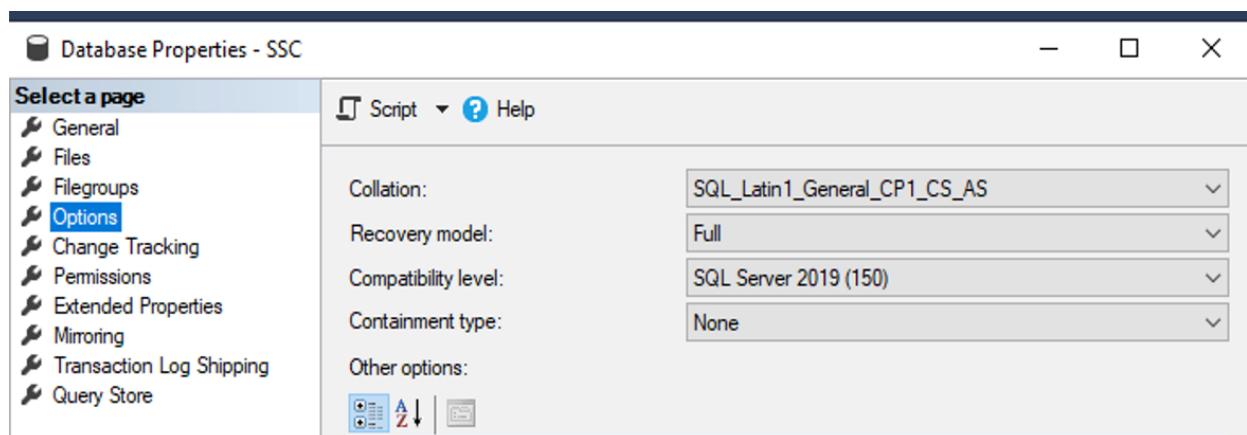
In certain cases, you can choose a parallel plan, even though the query's cost plan is less than the current cost threshold for parallelism value. This can happen if the decision to use a parallel or serial plan is based on a cost estimate provided earlier in the optimization process. For information about the cost threshold for parallelism option, see <https://www.brentozar.com/archive/2017/03/why-cost-threshold-for-parallelism-shouldnt-be-set-to-5>.

It is critical that Fortify Software Security Center users verify that the Compatibility level setting matches the current SQL Server engine version. Typically, when users upgrade to a newer SQL version, they back up and restore the Fortify Software Security Center database to a new SQL server that is running the latest supported SQL version. When you restore the Fortify Software Security Center database, the compatibility level is set to the SQL version on which the Fortify Software Security Center database was previously based.

The following screen shot shows an example of the properties for a Fortify Software Security Center database that was restored to MS SQL Server 2019.



The **Compatibility level** setting must be changed to reflect the current SQL Server engine version.



Sizing Recommendations for MS SQL Databases

The following recommendations are based on a limited data set and must not be construed as definitive. Their usability should increase as more data becomes available.

	Small	Medium	Large	X-Large
Data Size (TB)	1	1 - 3	3 - 5	5 - 10
Scans per Week	up to 5K	5 - 15 K	15 - 25 K	25+ K
Number of Users	up to 1 K	1 - 5 K	5 - 10 K	10+ K
Number of App Versions	up to 5 K	5 - 20 K	20 - 50 K	50+ K
RAM (GB)	64	128	256	512
Processors	8-core	16-core	32-core	64-core
IOPS - Minimum	32,000	32,000+	64,000	64,000+
*Cost Threshold for Parallelism	50	50	50	50

* The value can be increased based on evaluated performance needs. The value must be at least 50.

Note: Once the database reaches 5 TB in size, OpenText recommends that you deploy multiple Fortify Software Security Center servers.

Sizing Recommendations for MySQL Community Edition Databases

Caution! MySQL Community Edition databases are not recommended for large enterprise implementations.

Note: Once the database reaches 500 GB in size, OpenText recommends that you use MySQL Enterprise Edition databases.

	Small	Medium	Large	X-Large
Data Size	1 TB	1 TB - 3 TB	3 TB - 5 TB	5TB - 10TB
Scans per Week	up to 5 K			
Number of Users	up to 1 K			
Number of App Versions	up to 5 K			
RAM (GB)	64			
Processors	8-core			
IOPS - Minimum	32,000			
Cost Threshold for Parallelism	50			

Sizing Recommendations for Oracle Databases

	Small	Medium	Large	X-Large
Data Size TB	1	1 - 3	3 - 5	5 - 10
Scans per Week	up to 5 K	5 - 15 K	15 - 25 K	25,000+
Number of Users	up to 1 K	1 - 5 K	5 - 10 K	10,000+
Number of App Versions	up to 5 K	5 - 20 K	20 - 50 K	50,000+
RAM (GB)	64	128	256	512
Processors	8-core	16-core	32-core	64-core
IOPS - Minimum	32,000	32,000+	64,000	64,000+

Note: Once the database reaches 5 TB in size, OpenText recommends deploying multiple Fortify Software Security Center servers.

Chapter 3: Database Performance Issues

Disk I/O

Disk I/O encompasses the input/output operations on a physical disk. In reading data from a file on a disk, the processor must wait for the file to be read (the same applies to writing data to a file). Fortify Software Security Center is a high I/O-intensive application, which affects performance

Performance begins to degrade once the Fortify Software Security Center database reaches a certain size. Monitoring overall read/write latency is especially important as data volume approaches 1 TB. At that point, adjustments to the database deployment are required.

Fortify Software Security Center application workloads tend to grow rapidly in total data, size of the active data set, and the compute needed to satisfy growing transaction requirements.

Note: Table cleanups that use purge or delete of artifacts or application versions might not result in actual reduction of database storage allocation until the database administrator re-optimizes the database. OpenText recommends regular monitoring and optimization of the Fortify Software Security Center databases.

Amazon RDS Storage Types

Amazon RDS provides three storage types:

- General Purpose SSD – General Purpose SSD volumes offer cost-effective storage that is ideal for a broad range of workloads running on medium-sized database instances. General Purpose storage is best suited for development and testing environments. For more information about General Purpose SSD storage, including the storage size ranges, see **General Purpose SSD storage** on the Amazon Relational Database Service Documentation website (https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Storage.html).
- Provisioned IOPS SSD – Provisioned IOPS storage is designed to meet the needs of I/O-intensive workloads, particularly database workloads, that require low I/O latency and consistent I/O throughput. Provisioned IOPS storage is best suited for production environments. For more information about Provisioned IOPS storage, including the storage size ranges, see **Provisioned IOPS SSD storage** on the Amazon Relational Database Service Documentation website (https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Storage.html).
- Magnetic – Amazon RDS supports magnetic storage for backward compatibility. Amazon recommends that you use General Purpose SSD or Provisioned IOPS SSD for any new storage needs. The maximum amount of storage allowed for database instances on magnetic storage is less than that of the other storage types. For more information, see **Magnetic storage** on the Amazon Relational Database Service Documentation website (https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Storage.html).

Oracle Databases: Automatic Workload Repository for Database Tuning

Oracle's Automatic Workload Repository (AWR) collects, processes, and maintains performance statistics for problem detection and self-tuning purposes. An AWR report includes data on database activity between two points in time. The reports have many sections, each of which contains a large amount of database performance data. You can use this information to compare statistics captured during a period of poor performance to a baseline, and then diagnose performance issues.

OpenText recommends that, before you reach out to Customer Support for help with a Fortify Software Security Center performance issue, you first generate an AWR report and have your Oracle DBA review any recommendations it provides.

For Oracle databases, you can run the Automatic Workload Repository (AWR) report, which provides recommended changes. For detailed information about AWR, see

<https://docs.oracle.com/database/121/RACAD/GUID-C3CD2DCE-38BD-46BA-BC32-7A28CAC9A7FD.htm#RACAD951>.

Example AWR report

WORKLOAD REPOSITORY report for

DB Name	DB Id	Uniqe Name	Role	Edition	Release	RAC	CDB
FORTIFYFP	2479940867	fortifyfp_mil	PRIMARY	EE	19.0.0.0.0	YES	NO

Instance	Inst Num	Startup Time	User Name	System Data Visible
fortifyfp1	1	24-Oct-21 14:45	SYS	YES

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
dcmipdbgdb251.edc.nam.gm.com	Linux x86 64-bit	56	28	2	754.48

	Snap Id	Snap Time	Sessions	Cursors/Session	Instances
Begin Snap:	43880	19-Nov-21 00:00:11	130	1.3	2
End Snap:	43904	20-Nov-21 00:00:26	128	1.2	2
Elapsed:		1,440.25 (mins)			
DB Time:		9,293.37 (mins)			

Report Summary

Top ADDM Findings by Average Active Sessions

Finding Name	Avg active sessions of the task	Percent active sessions of finding	Task Name	Begin Snap Time	End Snap Time
Top SQL Statements	9.00	97.76	ADDM:2479940867_1_43892	19-Nov-21 11:00	19-Nov-21 12:00
Top SQL Statements	7.95	96.92	ADDM:2479940867_1_43897	19-Nov-21 16:00	19-Nov-21 17:00
Top SQL Statements	7.97	95.68	ADDM:2479940867_1_43891	19-Nov-21 10:00	19-Nov-21 11:00
Row Lock Waits	9.00	55.59	ADDM:2479940867_1_43892	19-Nov-21 11:00	19-Nov-21 12:00
Row Lock Waits	7.97	62.72	ADDM:2479940867_1_43891	19-Nov-21 10:00	19-Nov-21 11:00

Load Profile

	Per Second	Per Transaction	Per Exec	Per Call
DB Time(s):	6.5	0.3	0.04	0.01
DB CPU(s):	0.2	0.0	0.00	0.00
Background CPU(s):	0.0	0.0	0.00	0.00
Redo size (bytes):	289,622.1	14,069.5		
Logical read (blocks):	18,742.9	910.5		
Block changes:	1,484.5	72.1		
Physical read (blocks):	3,147.8	152.9		
Physical write (blocks):	21.6	1.1		
Read IO requests:	1,417.4	68.9		
Write IO requests:	10.4	0.5		
Read IO (MB):	24.6	1.2		
Write IO (MB):	0.2	0.0		
IM scan rows:	0.0	0.0		
Session Logical Read IM:	0.0	0.0		
Global Cache blocks received:	16.2	0.8		
Global Cache blocks served:	8.6	0.4		
User calls:	518.8	25.2		
Parse (SQL):	138.6	6.7		

MySQL Databases: Percona Toolkit

Percona offers a toolkit that includes the `pt-diskstats` tool, which you can use to monitor disk I/O for MySQL databases. The `pt-diskstats` tool is similar to `iostat` but is more interactive and detailed. For more information on the `pt-diskstats` tool, see <https://docs.percona.com/percona-toolkit/pt-diskstats.html>.

For MySQL databases, you can use MySQL Enterprise Monitor (<https://www.mysql.com/products/enterprise/monitor.html>) to help analyze and tune your database performance.

MySQL InnoDB Buffer Pool

MySQL InnoDB Buffer Pool is an in-memory cache between InnoDB pages on disk and MySQL query runs. When MySQL reads a page, it first checks if the page exists in the InnoDB buffer pool. MySQL skips the disk I/O operation and reads the pages from the buffer pool. If the pages do not exist in the buffer pool, MySQL retrieves the pages from the disk, adds the pages to the buffer pool, and then continues to run the query.

The buffer pool enhances the performance of MySQL query runs. Otherwise, a query overload uses more disk I/O operations.

Note: OpenText recommends that you should set the MySQL InnoDB Buffer Pool to $\frac{3}{4}$ of total memory.

Microsoft SQL Server

Customer support provides several SQL queries that you can use to gather data directly from MS SQL. Dynamic Management Views are used.

Dynamic management views and functions return server state information that you can use to monitor the health of a server instance, diagnose problems, and tune performance. (To run these queries, you must have SQL Server sysadmin rights.)

Note: OpenText recommends that you follow Microsoft SQL best practices.

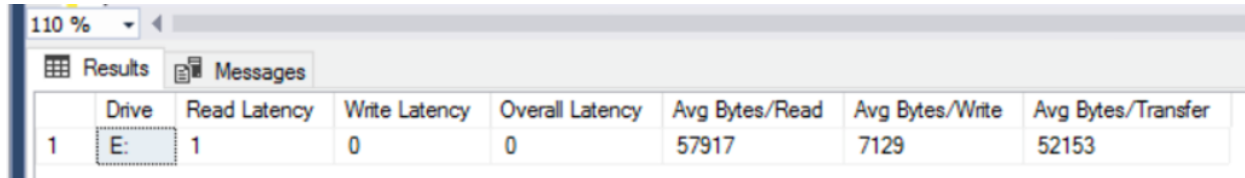
SQL Server Wait Statistics

One of the most under-used performance troubleshooting methodologies in the SQL Server world is *Wait Statistics*, or simply *wait stats*. The wait stats performance object contains performance counters that report information about broad categorizations of waits. You can identify the performance issue by analyzing the wait stats, and then use the results to see where to take the necessary actions to resolve the issue.

SQL Server "knows" where performance issues exist. One such issue is CPU Pressure. Signal waits above 10-15% often indicate a CPU pressure issue.

Note: For detailed information about Wait Statistics, see "SQL Server, Wait Statistics object" in Microsoft Learn (<https://learn.microsoft.com/en-us/sql/relational-databases/performance-monitor/sql-server-wait-statistics-object?view=sql-server-ver16>).

The following results, returned for a query for read/write latency (see "Query 7: Drive-level latency information" on page 40) shows results for the E:\ drive, which is where the Fortify Software Security Center database is located.



	Drive	Read Latency	Write Latency	Overall Latency	Avg Bytes/Read	Avg Bytes/Write	Avg Bytes/Transfer
1	E:	1	0	0	57917	7129	52153

The following table is a good reference against which to compare your read/write latency values.

Latency Value / Value Range	Interpretation
> 1 ms	Excellent
> 5 ms	Very good
5 - 10 ms	Good
10 - 20 ms	Poor
20 - 100 ms	Bad
100 - 500	Very bad
> 500 ms	Ridiculously bad

Checks for a Microsoft SQL Server Database

If you are using a SQL Server database as the Fortify Software Security Center database, perform the following checks:

- Enable the Auto Update Stats Asynchronously (AUTO_UPDATE_STATISTICS_ASYNC) option for the database. For instructions, see the Microsoft SQL documentation website (<https://docs.microsoft.com/en-us/sql/?view=sql-server-ver15>).
- Make sure that your SQL Server database schema collation is case-sensitive. The default installation of SQL Server is case-insensitive.

Caution! Fortify Software Security Center requires that all database schema collations be case-sensitive. If your database schema collation is case-insensitive, Fortify Software Security Center does not work correctly.

Important! Before you run the Fortify-provided SQL scripts, verify that there are no open connections to the database.

- Make sure that snapshot isolation is enabled (`ALLOW_SNAPSHOT_ISOLATION` and `READ_COMMITTED_SNAPSHOT` are set to ON) on the database schema used for the installation.
- During SQL script executions, check the client tool to make sure that its ANSI null default option is set to ON. To do this, you can either use a SET command (set `ANSI_NULL_DFLT_ON` to ON) or the Query Editor.

Indexing Fragmentation

Index fragmentation is a common source of database performance degradation. Fragmentation occurs when there is a lot of empty space on a data page (internal fragmentation) or when the logical order of pages in the index does not match the physical order of pages in the data file (external fragmentation).

Microsoft has its own maintenance solution, which is `AdaptiveIndexDefrag`. `AdaptiveIndexDefrag` performs an intelligent defrag on one or more indexes, for one or more databases. For details on what `AdaptiveIndexDefrag` does and how to use it, see "Adaptive Index Defrag" (<https://techcommunity.microsoft.com/t5/sql-server-blog/adaptive-index-defrag/ba-p/383893>).

Alternatively, you can use the SQL Server Index and Statistics Maintenance Solution. For details about this solution and how to use it, see "SQL Server Index and Statistics Maintenance" (<https://ola.hallengren.com/sql-server-index-and-statistics-maintenance.html>).

For its databases, Oracle provides a script to check for index fragmentation (<http://www.oracle-wiki.net/startscriptcheckfrag>). Be aware, however, that running this script takes locks out of the indexes.

For information on how to check for index fragmentation in MySQL databases, see the topic Check for Fragmentation in MySQL, and fix it (<https://aodba.com/check-fragmentation-mysql-fix>).

Fortify Software Security Center Scheduler

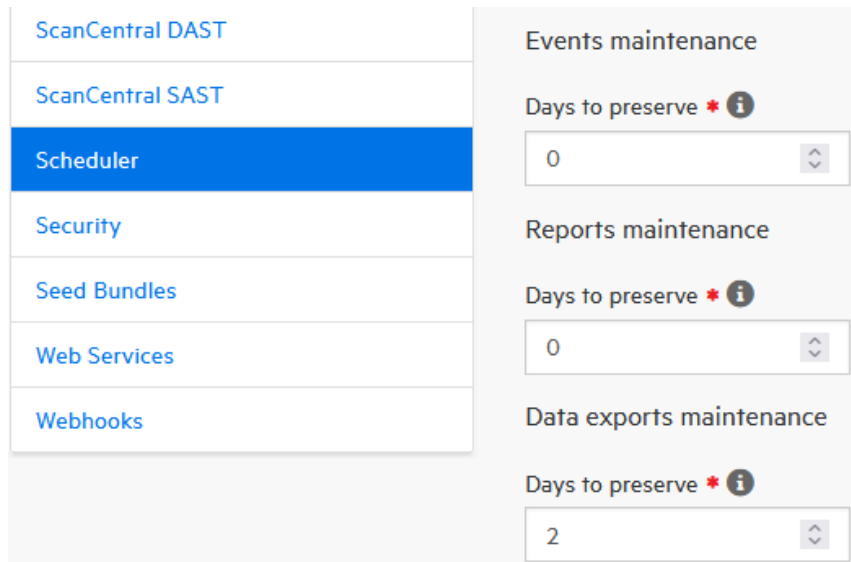
With the release of Fortify Software Security Center version 24.2.0, Fortify Software Security Center has a data retention policy that allows you to define the time period for which artifacts are retained in Fortify Software Security Center. You can configure advanced options to define the time period to retain the artifacts and the number of artifacts to retain per application version.

After the defined retention time period is reached, the artifacts are eligible for purging from Fortify Software Security Center. You can schedule the cleanup service to purge artifacts from Fortify Software Security Center. For more information, see *About Data Retention* in the *Fortify Software Security Center User Guide*.

If you allow artifacts and application versions to grow over time without maintenance, you may begin to see performance issues and extended upgrade times between releases. Users typically notice the degraded performance when their Fortify Software Security Center database reaches 1TB in size. This can easily happen if you have many years of unpurged data.

If you have not upgraded to Fortify Software Security Center version 24.2.0, Customer Support provides a set of PowerShell scripts that you can use to purge or delete artifacts. These scripts enable you to download the artifacts before you purge or delete them. The goal is to keep the database as trim and performant as possible.

Along with the data retention policy, Fortify Software Security Center provides the following three data retention settings (under **ADMINISTRATION > Configuration > Scheduler**):



- Events maintenance

This option enables you to specify the number of days after which Fortify Software Security Center removes past events. The default is zero (0), which results in no event removal.

Consider setting **Events Maintenance > Days to preserve** to 35. Anything higher could result in the addition of millions of rows to the `dbo.eventlogentry` table, which stores all events that you see in the Fortify Software Security Center user interface.

Date	Username	Notes	Type
07/13/2023 1:13:41 PM	gpatel	[Security Event]	User Logged In Successfully
07/13/2023 1:11:35 PM	jdu	[Security Event]	User Logged In Successfully
07/13/2023 10:23:29 AM	amit	[Security Event]	User Logged Out
07/13/2023 10:23:29 AM	amit		Session Timeout
07/13/2023 9:34:23 AM	amit	[Security Event]	User Logged In Successfully
07/13/2023 9:05:00 AM	System	[Security Event]	LDAP Cache Refresh Started

You can use the **EXPORT** button on the Event Logs page in Fortify Software Security Center (**ADMINISTRATION > Metrics & Tracking > Event Logs**) to back up all existing events. After you do, you can safely truncate the `dbo.eventlogentry` table. OpenText recommends that you truncate the table on a regular basis.

- Reports maintenance

You can manage the reports generated on your Fortify Software Security Center instance. Users (with required permissions) on Fortify Software Security Center versions earlier than 21.1.x can delete generated reports manually. In 22.1.0 and later versions, you can set **Reports maintenance > Days to preserve** so reports that are no longer required are deleted automatically after the specified number of days.

If you are not yet using Fortify Software Security Center 22.1.x or later version, Customer Support can provide a PowerShell script that you can run to create a list of generated reports, which you can then use to delete reports no longer required.

- Data exports maintenance

When users perform a data export in Fortify Software Security Center, the exported data are stored. By default, **Data exports maintenance > Days to preserve** is set to 2. OpenText recommends that you leave the default setting of 2 to ensure that exported data are regularly removed from the database.

Managing Authentication Tokens

There is no data retention policy for expired tokens in Fortify Software Security Center. OpenText recommends that administrators periodically review the list of authentication tokens generated from the Fortify Software Security Center user interface or from the command-line interface, and delete the tokens that have expired. For information about how to delete authentication tokens, see the *OpenText™ Fortify Software Security Center User Guide*.











Managing Artifacts

An artifact in Fortify Software Security Center is a container for various types of content. The most important artifacts are Fortify project results files, or FPRs. An FPR usually contains one scan produced by a specific analyzer (for example, OpenText™ Fortify Static Code Analyzer or OpenText™ Fortify WebInspect), but can include multiple scans produced by different analyzers. When FPRs are uploaded to Fortify Software Security Center, the corresponding artifact gets a system date stamp as the artifact upload date. A scan inside of an artifact has a scan date apart from the artifact upload date.

For simplicity sake, assume that:

- Each of several artifacts uploaded to Fortify Software Security Center contains just one FPR file
- The artifacts were uploaded in the same order that the scans that produced the results were performed. So, for example, the FPR resulting from the oldest scan is uploaded first.

After all artifacts are uploaded, the **ARTIFACT HISTORY** list looks as follows, where v1 . fpr contains the oldest scan data, and v6 . fpr contains the newest.

ARTIFACT HISTORY						
 ARTIFACT		 APPLICATION FILE		 APPLICATION & SOURCES		 REFRESH
Upload Date	Status	Uploaded By	Type	Audits	Scan Artifact	
11/30/2021 6:07:57 PM	Complete	admin	SCA		gold_openme ... 5.2.2_v6.fpr	
11/30/2021 6:07:18 PM	Complete	admin	SCA		gold_openme ... 5.2.2_v5.fpr	
11/30/2021 6:07:17 PM	Complete	admin	SCA		gold_openme ... 5.2.2_v4.fpr	
11/30/2021 6:07:15 PM	Purged	admin	SCA		gold_openme ... 5.2.2_v3.fpr	
11/30/2021 6:07:13 PM	Purged	admin	SCA		gold_openme ... 5.2.2_v2.fpr	
11/30/2021 6:07:11 PM	Purged	admin	SCA		gold_openme ... 5.2.2_v1.fpr	

Artifact Q&A

Q: On a brand new Fortify Software Security Center version, I can add multiple Fortify Static Code Analyzer artifacts, and then delete any of them. How does the state handle this?

A: When you delete any of the artifacts, Fortify Software Security Center recalculates the new state based on the remaining artifacts. When the last artifact associated with an application version is deleted, the application version acquires “no state” (it becomes a completely empty application version).

Q: If I have five artifacts and I purge numbers 6 through 10, why is it possible to delete numbers 1 through 4, but not 5? That makes no sense to me.

A: Issues are stored in two places, one for scan issues storage and one for issue storage. *Scan issue storage* holds issues associated with each scan uploaded to Fortify Software Security Center. This storage contains only issues present in a specific scan. *Issue storage* contains the current issues state for each application version.

The following example illustrates the difference:

Scan 1 contains issue A and Scan 2 contains issues A and B.











In this example, *Scan issue storage* contains three issues: one for Scan 1 (issue A) and two for Scan 2 (issues A and B). *Issue storage* contains two issues: A (the status of which is UPDATED) and B (the status of which is NEW).

This separation is needed to simplify and speed up the querying of current issues state. Fortify Software Security Center does not need to analyze all the scan issues to calculate issue status at the moment of query. It just selects precalculated current issues from issue storage.

The separation would be unnecessary if Fortify Software Security Center used the latest scan result as the issue state. But, in addition to tracking when the issue was updated (found by the current and previous scans), Fortify Software Security Center also tracks removed and reintroduced issues. Because of this, having scan issues for previous scans is important for current scan processing and final issue state calculation.

Issue storage state must be recalculated every time a new scan is uploaded to Fortify Software Security Center. Scan issue storage is the primary source of information used for issue state calculation.

A purge operation removes data from scan issue storage to decrease the amount of disc space the Fortify Software Security Center database uses. Note that purging *does not* delete data from current issues storage and does not affect the current application version issues state. So, if you purged v3.fpr, you would see the following:

ARTIFACT HISTORY						
 ARTIFACT		 APPLICATION FILE		 APPLICATION & SOURCES		 REFRESH
Upload Date	Status	Uploaded By	Type	Audits	Scan Artifact	
11/30/2021 6:07:57 PM	Complete	admin	SCA		gold_openme ... 5.2.2_v6.fpr	
11/30/2021 6:07:18 PM	Complete	admin	SCA		gold_openme ... 5.2.2_v5.fpr	
11/30/2021 6:07:17 PM	Complete	admin	SCA		gold_openme ... 5.2.2_v4.fpr	
11/30/2021 6:07:15 PM	Purged	admin	SCA		gold_openme ... 5.2.2_v3.fpr	
11/30/2021 6:07:13 PM	Purged	admin	SCA		gold_openme ... 5.2.2_v2.fpr	
11/30/2021 6:07:11 PM	Purged	admin	SCA		gold_openme ... 5.2.2_v1.fpr	

Fortify Software Security Center now contains full scan data related to scans 6, 5, and 4 (scan and scan issues). Scan issues associated with scans 3, 2 and 1 are removed. Once that happens, Fortify Software Security Center cannot allow the deletion of any purged scans because scan issues associated with these scans are gone and Fortify Software Security Center cannot reliably recalculate the state of all issues whose status is not NEW, but REMOVED, REINTRODUCED and UPDATED.

Notes on artifact/scan ordering

Even if you can upload artifacts in an order that is different from the order of scan dates, *OpenText strongly recommends that not to do it*, especially if you want to use purging. The visual representation that would result in the ARTIFACT HISTORY table would be confusing.

The ARTIFACT HISTORY table, with its fixed upload time order, is a bit of a limiting feature of Fortify Software Security Center. A scan date column cannot currently be added to the table because of the 1:N relation between FPRs and scans.

Q: Why is the analysis date used to determine the issues shown (instead of the upload date)?

A: This is the only reliable way to track issue status. If the upload date was used to calculate the current issues state, the results would be inconsistent.

To illustrate:

Scan 1 uncovered one issue in the application source code. This issue was fixed, and scan 2 uncovered no issues. The order in which the results of these two scans are uploaded does not matter. The issue status is always removed once both scans are uploaded. If Fortify Software Security Center used the upload date for scan calculation, it would not be true. If scan 1 was uploaded before scan 2, the issue status would be REMOVED. But if scan 2 is uploaded first, followed by scan 1, the issue status would be incorrectly set to NEW, which is wrong. The issue that no longer exists in code would be marked as an active, new issue.

Artifact Delete/Purge Scripts

Customer Support provides the following set of PowerShell scripts that you can use to either purge or delete artifacts.

- `configFile.ps1`
- `GenerateListofArtifacts.ps1`
- `DeleteArtifacts.ps1`
- `PurgeArtifacts.ps1`
- `ConnectivityTestToSSC.ps1`

These scripts enable you to download artifacts and then perform the purge or delete so that the database is kept as trim as possible. If you use the scripts to download the artifacts, they create a parent directory based on the application name, and a sub-directory based on the application version. The downloaded artifacts are placed in the output directory that you specify in the `configFile.ps1` script.

The `ConnectivityTestToSSC.ps1`, which you run first, is designed to test connectivity between the machine from which you run the PowerShell scripts and Fortify Software Security Center. The script attempts to connect to Fortify Software Security Center, then requests a Unified Token.

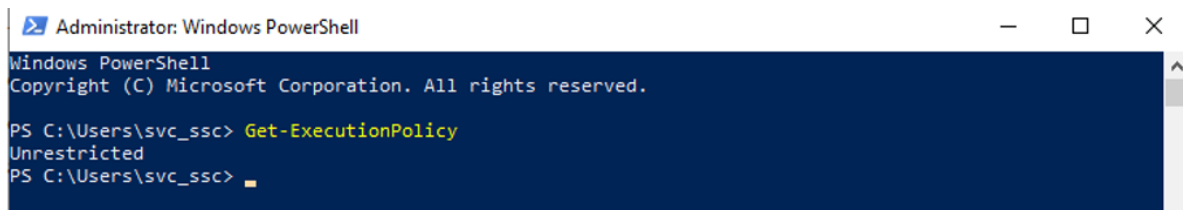
Before you run any of the PowerShell scripts, you must first verify the PowerShell Execution Policy on Windows and connect to Fortify Software Security Center:

1. Run the following command using Windows PowerShell ISE invoked as administrator:

```
get-executionpolicy
```

To set the policy to unrestricted, run:

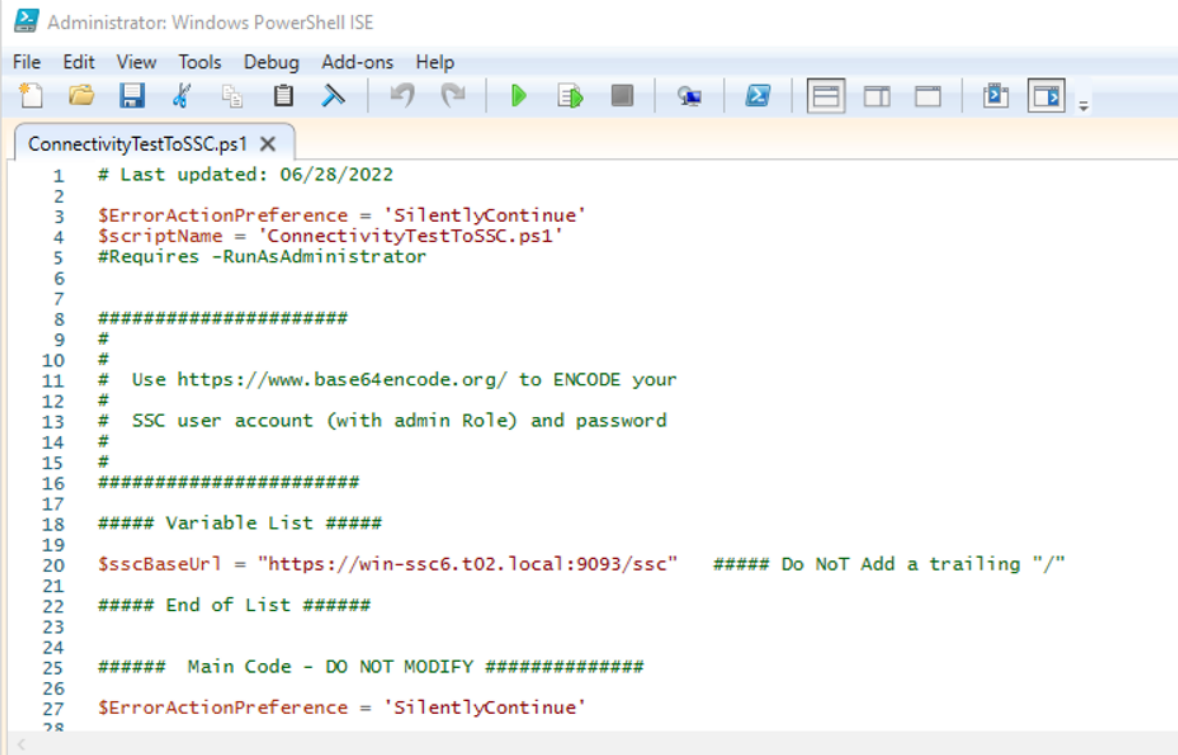
```
set-executionpolicy unrestricted
```



The screenshot shows a Windows PowerShell ISE window titled "Administrator: Windows PowerShell". The window content displays the following text:

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
PS C:\Users\svc_ssc> Get-ExecutionPolicy  
Unrestricted  
PS C:\Users\svc_ssc> █
```

- Use PowerShell ISE (run as administrator) to run the ConnectivityTestToSSC.ps1 script, as shown here:

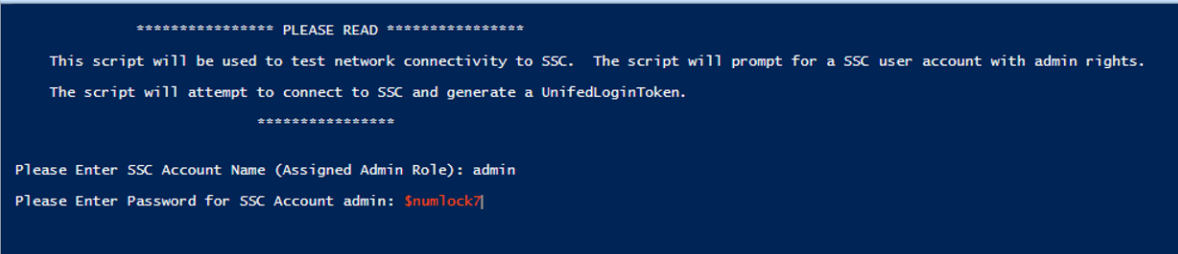


```

Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
ConnectivityTestToSSC.ps1 X
1 # Last updated: 06/28/2022
2
3 $ErrorActionPreference = 'SilentlyContinue'
4 $scriptName = 'ConnectivityTestToSSC.ps1'
5 #Requires -RunAsAdministrator
6
7
8 #####
9 #
10 #
11 # Use https://www.base64encode.org/ to ENCODE your
12 #
13 # SSC user account (with admin Role) and password
14 #
15 #
16 #####
17
18 ##### Variable List #####
19
20 $sscBaseUrl = "https://win-ssc6.t02.local:9093/ssc" ##### Do NOT Add a trailing "/"
21
22 ##### End of List #####
23
24
25 ##### Main Code - DO NOT MODIFY #####
26
27 $ErrorActionPreference = 'SilentlyContinue'
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

- Provide credentials when prompted. (The information you enter is not saved.)

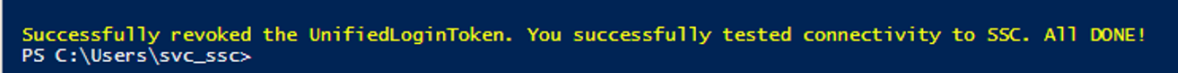


```

***** PLEASE READ *****
This script will be used to test network connectivity to SSC. The script will prompt for a SSC user account with admin rights.
The script will attempt to connect to SSC and generate a UnifiedLoginToken.
*****
Please Enter SSC Account Name (Assigned Admin Role): admin
Please Enter Password for SSC Account admin: $numlock7

```

After the script successfully connects to Fortify Software Security Center, it directs Fortify Software Security Center to create a token, and then revoke that token.



```

Successfully revoked the UnifiedLoginToken. You successfully tested connectivity to SSC. ALL DONE!
PS C:\Users\svc_ssc>

```


Example:

```

ArtifactsDeleteList.txt
1 "Total Number Of Artifacts","ProjectName","VersionName","ProjectID","ArtifactID","allowDelete","Upload_Date","Status","ArtifactFileName","FileSize","Comment"
2 "3","DeletePurgeTest","1.0","10038","740","True","7/6/2022 9:48:17 AM","PROCESS_COMPLETE","webgoat_1.fpr","857587",""
3 "3","DeletePurgeTest","1.0","10038","742","True","7/6/2022 9:48:17 AM","PROCESS_COMPLETE","webgoat_2.fpr","1028014",""
4 "3","DeletePurgeTest","1.0","10038","744","True","7/6/2022 9:48:18 AM","PROCESS_COMPLETE","webgoat_3.fpr","1970337",""
5 "5","Test-Scripts-Artifacts","1.0","10043","750","True","7/7/2022 5:21:33 PM","PROCESS_COMPLETE","webgoat_3.fpr","1970337",""
6 "5","Test-Scripts-Artifacts","1.0","10043","752","True","7/7/2022 5:21:33 PM","PROCESS_COMPLETE","webgoat_4.fpr","1983214",""
7 "5","Test-Scripts-Artifacts","1.0","10043","754","True","7/8/2022 8:50:15 AM","REQUIRE_AUTH","webgoat_1.fpr","857587",""
8 "5","Test-Scripts-Artifacts","1.0","10043","756","True","7/8/2022 8:50:15 AM","PROCESS_COMPLETE","webgoat_2.fpr","1028014",""
9 "5","Test-Scripts-Artifacts","1.0","10043","758","True","7/8/2022 8:50:15 AM","PROCESS_COMPLETE","webgoat_3.fpr","1970337",""
10
    
```

- SSCProjectsWithNoArtifacts.txt lists Fortify Software Security Center application versions that have no associated artifacts.

The href can be used to submit a POST Delete to delete each application version with no associated artifacts.

Example:

```

SSCProjectsWithNoArtifacts.txt
1 "href","ApplicationName","versionName","ProjectID"
2 "https://win-ssc6.t02.local:9093/ssc/api/v1/projectVersions/10044","AppTesting123","1.0","10044"
3 "https://win-ssc6.t02.local:9093/ssc/api/v1/projectVersions/10042","DeletePurgeTest","2.0","10042"
4 "https://win-ssc6.t02.local:9093/ssc/api/v1/projectVersions/10041","NoArtifactsInApplicationVersion","1.0","10041"
5
    
```

- ArtifactsWithPurging_Status.txt lists Fortify Software Security Center artifacts with the "Purging" status.

Example:

```

artifactsWithPURGING_Status.txt
1 The artifact with an ID of 760 (upload DATE: 07/13/2022 11:26:37 | file Name: webgoat_1.fpr) associated with the AppTesting123/1.0 Application has a status of 'PURGING'.
2
    
```

If you have artifacts that are stuck in 'Purging' status, contact Customer Support for assistance.

- SummaryReport_Deletes.txt file displays the number of artifacts that can be deleted, and the number that would remain after deletion is completed.

Example:

```

SummaryReport_DELETES.txt
1 Summary Report: DeletePurgeTest (10038) 1.0 | Total Number of Artifacts: 1 | Total Number of Deletable Artifacts: 1 | Remaining Artifacts: 0
2 Summary Report: Test-Scripts-Artifacts (10043) 1.0 | Total Number of Artifacts: 5 | Total Number of Deletable Artifacts: 5 | Remaining Artifacts: 0
3
    
```

If you decide to delete artifacts, this file gives you an idea of how many artifacts could be deleted and how many would be left. If, after deletion, an application version would have zero artifacts, you can just delete it from the ArtifactsDeleteList.txt file.

Deleting artifacts

To delete artifacts:

1. Run the DeleteArtifacts.ps1 script.

The following information is displayed:

```

***** IMPORTANT *****
Based on the date provided (when you inolved the GenerateListofArtifacts.ps1), artifacts have
been found that could be DELETED.
Please review the file 'D:/Powershell script/Purge_Delete
    
```

```

Scripts/output/ArtifactsDeleteList.txt'.
The generated file will be used by the DeleteArtifacts.ps1 script.
****Delete operation will revert all traces of an artifact!****
****Application history will be impacted!****
RECOMMENDATION: If you are planning to DELETE the artifacts no need to download.
The Delete script will prompt you if you wish to download the artifacts.
RECOMMENDATION: DELETE in batches.
*****

```

The delete operation removes all traces of an artifact. Application history is affected, but you free up more storage area in the Fortify Software Security Center database.

Caution! For every purge request, a corresponding purge job is created in Fortify Software Security Center. OpenText strongly recommends that you purge artifacts in batches of no more than 100 at a time. You can monitor the corresponding Delete jobs to track performance and potentially increase the number of artifacts per batch.

2. Respond to the following prompts. You can tell the script to download the artifacts before their deletion. (The credentials you provide used to access Fortify Software Security Center are not saved.)

```

Did you READ the above Statements (Yes)??:
The user has acknowledged that the statements have been reviewed.
Confirming that you wish to DELETE the artifacts (Yes)??:
The user has acknowledged that they wish to DELETE the artifacts (listed in 'D:/Powershell
script/Purge_Delete Scripts/output/ArtifactsDeleteList.txt'). An SSC Artifact 'Delete job'
will be created for each artifact. You can exit the script now to change your option.
Artifacts with status of 'ERROR PROCESSING' will be ignored for downloading. Download
Artifacts before the 'DELETE' operation occurs (Yes/No)??:
The user has acknowledged that the artifact(s) will be downloaded prior to the 'DELETE'
operation.
Please Enter SSC Account Name (Assigned Admin Role):
Please Enter Password for SSC Account admin:

```

Purging artifacts

The purge operation removes artifacts from the system and recovers space in the database without affecting issue metrics. The database space reclaimed is not as extensive as that reclaimed by the delete artifact operation.

To purge artifacts:

1. Run the `PurgeArtifacts.ps1` script.
The following Information is displayed:

```

***** IMPORTANT *****
Based on the date provided (when you invoked the GenerateListofArtifacts.ps1), artifacts have
been found that could be PURGED.
Please review the file 'D:/Powershell script/Purge_Delete
Scripts/output/ArtifactsPurgeList.txt'.
The generated file will be used by the PurgeArtifacts.ps1 script.
    ****Purging artifacts from the system recovers space without affecting issue metrics****
    ****Application history will be impacted!****
RECOMMENDATION: If you are planning to PURGE the artifacts we recommend that you download
them.
The Purge script will prompt you if you wish to download the artifacts.
RECOMMENDATION: PURGE in batches.
*****

```

Caution! For every purge request there is a corresponding purge job created in Fortify Software Security Center. OpenText strongly recommends that you purge artifacts in batches of no more than 100 at a time. You can monitor the corresponding Delete jobs to track performance and potentially increase the number of artifacts per batch.

2. Respond to the prompts as the following content is displayed. (You can direct the script to download the artifacts before deleting them. The account and password used to access Fortify Software Security Center are not saved.)

```

Did you READ the above Statements (Yes)?:
The user has acknowledged that the statements have been reviewed.
Confirming that you wish to PURGE the artifacts (Yes)?:
The user has acknowledged that they wish to purge the artifacts (listed in'D:/Powershell
script/ Purge_Delete Scripts/output/ArtifactsPurgeList.txt'). An SSCArtifact 'Purge job' will
be created for each artifact. You can exit the script now to change your option.
Artifacts with the Status of 'ERROR PROCESSING' or 'REQUIRE AUTH' will be ignored for
downloading. Download Artifacts before the 'PURGE' operation occurs (Yes/No)?:
The user has acknowledged that the artifact(s) will be downloaded prior to the 'PURGE'
operation.
Please Enter SSC Account Name (Assigned Admin Role):
Please Enter Password for SSC Account admin:

```

Tables for Removing Data from the Fortify Software Security Center Database

If you are not actively using the Fortify Software Security Center Dashboard, you can delete data from the following Fortify Software Security Center database tables to free up space in your database. (You will need to shrink the database to reduce its overall size.)

- snapshotquickvalues
- variablehistory
- measurementhistory
- snapshot

You can use several queries to safely remove all data from the four tables. Once you run the queries, Fortify Software Security Center clears all trend data and application version metrics. It also clears the data from the Dashboard view in the user interface.

To remove data from the snapshotquickvalues, variablehistory, measurementhistory, and snapshot tables, use the following commands:

```
delete from snapshotquickvalues
```

```
delete from variablehistory
```

```
delete from measurementhistory
```

```
delete from snapshot
```

After the data are removed from the four tables, the Fortify Software Security Center Dashboard displays no data. As new artifacts are uploaded to Fortify Software Security Center and processed, the Dashboard displays the new data.

As an alternative to deleting data from the four tables directly, you can simply delete data associated with old snapshots. In this case, you must modify all the queries to use IDs from the snapshot table to be used in the “NOT IN” SQL expression, as follows:

```
delete from snapshotquickvalues
where snapshot_id not in (select latestSnapshot_id from projectversion)
delete from variablehistory
where snapshot_id not in (select latestSnapshot_id from projectversion)
delete from measurementhistory
where snapshot_id not in (select latestSnapshot_id from projectversion)
delete from snapshot
where id not in (select latestSnapshot_id from projectversion)
```

Note: OpenText recommends that you consult with your database administrators when you want to delete data from the four tables. In case, you are attempting to delete data from a table that

contains millions of rows, OpenText recommends to delete the data in batches and ensure your transaction log has sufficient space.

Maintenance Schedule

Database versus Data Warehouse

A *database* is designed to make transactional systems run efficiently. Typically, it is an online transaction processing (OLTP) database, which is usually constrained to a single application.

A *data warehouse* is a database of a different kind. A data warehouse exists as a layer on top of another database or databases (usually OLTP databases). The data warehouse takes the data from all these databases and creates a layer optimized for and dedicated to analytics. Fortify Software Security Center is not designed as a data warehouse.

Keeping 10 years' worth of data in Fortify Software Security Center is not practical unless you deploy more than one Fortify Software Security Center instance. If deploying multiple Fortify Software Security Center instances is not an option, OpenText recommends that you keep no more than two years of data for optimal performance.

For an application version that is no longer active or needed, you can download the latest merged scan results. (For instructions, see the *Fortify Software Security Center User Guide*.) Once the data are downloaded, you can delete the application version. If you need to access the information in the FPR, upload it to a non-production Fortify Software Security Center server.

Purge/Delete Script Schedule

OpenText recommends that you run the analysis, purge, and delete queries every six months, or sooner if you observe performance issues. If your database is over 1TB, OpenText recommends that you analyze and clean your database quarterly.

If you have several years of data, and you want to maintain only two years of data, you can use the purge script or delete script to purge or delete older artifacts.

The delete operation removes all traces of an artifact and the application version history is affected, and you reclaim storage space in the Fortify Software Security Center database. The purge operation also enables you to reclaim space in the database, but less than the delete operation, since issue history is maintained.

Appendix A: Database Queries: MS SQL (On Prem)

This section provides lists of queries for MS SQL (on prem) databases. These queries use the dynamic management views shipped with MS SQL. OpenText recommends that your database administrator execute these queries.

If you are a new Fortify Software Security Center user, you must run these queries to establish a baseline. As your database grows, and approaches 1 TB in size, consider re-running the queries and comparing the data to the baseline data.

If you are an experienced Fortify Software Security Center user, and you are seeing performance issues, use these queries to collect the necessary data so that Customer Support can use it to provide feedback and recommendations.

An output example and a description of what to look for are provided for each query.

Note: Queries for MySQL and Oracle database types, if applicable, are to be added in a future release.

Query 1: Listing SQL wait types

The following query, run against the master database, generates a list of SQL wait types:

```
IF OBJECT_ID('tempdb..#ignorable_waits') IS NOT NULL
DROP TABLE #ignorable_waits;
GO
create table #ignorable_waits (wait_type nvarchar(256) PRIMARY KEY);
GO
/* We aren't using row constructors to be SQL 2005 compatible */
set nocount on;
insert #ignorable_waits (wait_type) VALUES ('REQUEST_FOR_DEADLOCK_SEARCH');
insert #ignorable_waits (wait_type) VALUES ('SQLTRACE_INCREMENTAL_FLUSH_SLEEP');
insert #ignorable_waits (wait_type) VALUES ('SQLTRACE_BUFFER_FLUSH');
insert #ignorable_waits (wait_type) VALUES ('LAZYWRITER_SLEEP');
insert #ignorable_waits (wait_type) VALUES ('XE_TIMER_EVENT');
insert #ignorable_waits (wait_type) VALUES ('XE_DISPATCHER_WAIT');
insert #ignorable_waits (wait_type) VALUES ('FT_IFTS_SCHEDULER_IDLE_WAIT');
insert #ignorable_waits (wait_type) VALUES ('LOGMGR_QUEUE');
insert #ignorable_waits (wait_type) VALUES ('CHECKPOINT_QUEUE');
insert #ignorable_waits (wait_type) VALUES ('BROKER_TO_FLUSH');
insert #ignorable_waits (wait_type) VALUES ('BROKER_TASK_STOP');
insert #ignorable_waits (wait_type) VALUES ('BROKER_EVENTHANDLER');
insert #ignorable_waits (wait_type) VALUES ('SLEEP_TASK');
```

```

insert #ignorable_waits (wait_type) VALUES ('WAITFOR');
insert #ignorable_waits (wait_type) VALUES ('DBMIRROR_DBM_MUTEX')
insert #ignorable_waits (wait_type) VALUES ('DBMIRROR_EVENTS_QUEUE')
insert #ignorable_waits (wait_type) VALUES ('DBMIRRORING_CMD');
insert #ignorable_waits (wait_type) VALUES ('DISPATCHER_QUEUE_SEMAPHORE');
insert #ignorable_waits (wait_type) VALUES ('BROKER_RECEIVE_WAITFOR');
insert #ignorable_waits (wait_type) VALUES ('CLR_AUTO_EVENT');
insert #ignorable_waits (wait_type) VALUES ('DIRTY_PAGE_POLL');
insert #ignorable_waits (wait_type) VALUES ('HADR_FILESTREAM_IOMGR_IOCOMPLETION');
insert #ignorable_waits (wait_type) VALUES ('ONDEMAND_TASK_QUEUE');
insert #ignorable_waits (wait_type) VALUES ('FT_IFSHC_MUTEX');
insert #ignorable_waits (wait_type) VALUES ('CLR_MANUAL_EVENT');
insert #ignorable_waits (wait_type) VALUES ('SP_SERVER_DIAGNOSTICS_SLEEP');
insert #ignorable_waits (wait_type) VALUES ('QDS_CLEANUP_STALE_QUERIES_TASK_MAIN_LOOP_SLEEP');
insert #ignorable_waits (wait_type) VALUES ('QDS_PERSIST_TASK_MAIN_LOOP_SLEEP');
GO
/* Want to manually exclude an event and recalculate?*/
/* insert #ignorable_waits (wait_type) VALUES (""); */
/*****
What are the highest overall waits since startup?
*****/
SELECT TOP 25
os.wait_type,
SUM(os.wait_time_ms) OVER (PARTITION BY os.wait_type) as sum_wait_time_ms,
CAST(
100.* SUM(os.wait_time_ms) OVER (PARTITION BY os.wait_type)
/ (1. * SUM(os.wait_time_ms) OVER () )
AS NUMERIC(12,1)) as pct_wait_time,
SUM(os.waiting_tasks_count) OVER (PARTITION BY os.wait_type) AS sum_waiting_tasks,
CASE WHEN SUM(os.waiting_tasks_count) OVER (PARTITION BY os.wait_type) > 0
THEN
CAST(
SUM(os.wait_time_ms) OVER (PARTITION BY os.wait_type)
/ (1. * SUM(os.waiting_tasks_count) OVER (PARTITION BY os.wait_type))
AS NUMERIC(12,1))
ELSE 0 END AS avg_wait_time_ms,
CURRENT_TIMESTAMP as sample_time
FROM sys.dm_os_wait_stats os
LEFT JOIN #ignorable_waits iw on
os.wait_type=iw.wait_type
WHERE

```



```
iw.wait_type is null
ORDER BY sum_wait_time_ms DESC;
GO
```

Example output (SQL waits will vary):

	wait_type	sum_wait_time_ms	pct_wait_time	sum_waiting_tasks	avg_wait_time_ms	sample_time
1	PARALLEL_REDO_WORKER_WAIT_WORK	9408	45.2	745	12.6	2022-08-16 11:30
2	PWAIT_ALL_COMPONENTS_INITIALIZED	2900	13.9	3	966.7	2022-08-16 11:30
3	LCK_M_S	2659	12.8	16	166.2	2022-08-16 11:30
4	WAIT_XTP_HOST_WAIT	1306	6.3	3	435.3	2022-08-16 11:30
5	PAGEIOLATCH_2SH	818	3.9	1011	0.8	2022-08-16 11:30
6	IO_COMPLETION	665	3.2	482	1.4	2022-08-16 11:30
7	PREEMPTIVE_OS_FILEOPS	652	3.1	237	2.8	2022-08-16 11:30
8	SLEEP_DBSTARTUP	354	1.7	6	59.0	2022-08-16 11:30

The following three SQL waits are useful for finding disk I/O bottlenecks and for making sure that the READ_COMMITTED_SNAPSHOT database option is enabled on the Fortify Software Security Center database:

- The *LCK_M_S* wait occurs if a request is waiting to acquire a shared lock. This typically happens when read requests are blocked by write transactions (implicit or explicit) that have been kept open for extended periods of time.
- The *LCK_M_X* wait occurs if a transaction is waiting to acquire an exclusive lock in order to modify data. This lock prevents other transactions from accessing the objects, so no other processes can read or modify data.

Lock waits commonly occur on busy servers where concurrent transactions demand the same resource, resulting in poor performance. A high number of locking waits may indicate blocking problems and should be investigated.

Note: If you see LCK_M_X, LCK_M_IX in the list, make sure that the READ_COMMITTED_SNAPSHOT option is enabled.

- The *LCK_M_U* wait occurs while a request is waiting to acquire an update lock. An update lock is not just for UPDATE operations. It is used when SQL Server needs to read, and then modify, a row, page, or table. Before SQL Server makes any changes, it places an update lock on the data. Once the system is ready, these locks are upgraded to exclusive locks. This wait typically occurs while modify requests are blocked by other write transactions (implicit or explicit).

Note: If you see CXPACKET in the list of SQL wait types, see the article "Why Cost Threshold For Parallelism Shouldn't Be Set To 5" (<https://www.brentozar.com/archive/2017/03/why-cost-threshold-for-parallelism-shouldnt-be-set-to-5>).

Query 2: Signal Waits (Run against the master DB)

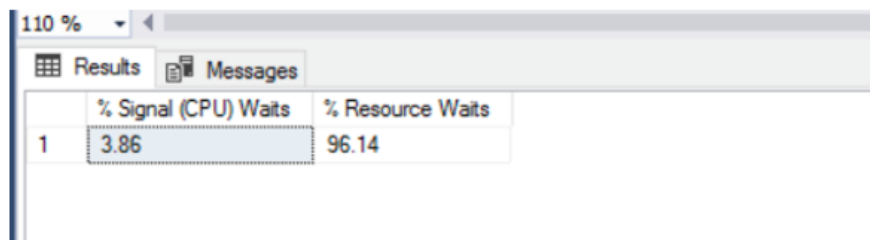
Signal waits indicate possible internal CPU pressure. The CPU signal waits percent is a ratiometric that compares signal waits to total waits, as a percent. That means you can see a spike in signal waits from one minute to the next without the server itself showing high CPU use.

```

SELECT CAST(100.0 * SUM(signal_wait_time_ms) / SUM (wait_time_ms) AS NUMERIC(20,2)) AS [% Signal
(CPU) Waits],
CAST(100.0 * SUM(wait_time_ms - signal_wait_time_ms) / SUM (wait_time_ms) AS NUMERIC(20,2)) AS [%
Resource Waits]
FROM sys.dm_os_wait_stats WITH (NOLOCK)
WHERE wait_type NOT IN (
    N'BROKER_EVENTHANDLER', N'BROKER_RECEIVE_WAITFOR', N'BROKER_TASK_STOP',
    N'BROKER_TO_FLUSH', N'BROKER_TRANSMITTER', N'CHECKPOINT_QUEUE',
    N'CHKPT', N'CLR_AUTO_EVENT', N'CLR_MANUAL_EVENT', N'CLR_SEMAPHORE',
    N'DBMIRROR_DBM_EVENT', N'DBMIRROR_EVENTS_QUEUE', N'DBMIRROR_WORKER_QUEUE',
    N'DBMIRRORING_CMD', N'DIRTY_PAGE_POLL', N'DISPATCHER_QUEUE_SEMAPHORE',
    N'EXECSYNC', N'FSAGENT', N'FT_IFTS_SCHEDULER_IDLE_WAIT', N'FT_IFTSHC_MUTEX',
    N'HADR_CLUSAPI_CALL', N'HADR_FILESTREAM_IOMGR_IOCOMPLETION', N'HADR_LOGCAPTURE_WAIT',
    N'HADR_NOTIFICATION_DEQUEUE', N'HADR_TIMER_TASK', N'HADR_WORK_QUEUE',
    N'KSOURCE_WAKEUP', N'LAZYWRITER_SLEEP', N'LOGMGR_QUEUE', N'ONDEMAND_TASK_QUEUE',
    N'PWAIT_ALL_COMPONENTS_INITIALIZED', N'QDS_PERSIST_TASK_MAIN_LOOP_SLEEP',
    N'QDS_CLEANUP_STALE_QUERIES_TASK_MAIN_LOOP_SLEEP', N'REQUEST_FOR_DEADLOCK_SEARCH',
    N'RESOURCE_QUEUE', N'SERVER_IDLE_CHECK', N'SLEEP_BPOOL_FLUSH', N'SLEEP_DBSTARTUP',
    N'SLEEP_DCOMSTARTUP', N'SLEEP_MASTERDBREADY', N'SLEEP_MASTERMDREADY',
    N'SLEEP_MASTERUPGRADED', N'SLEEP_MSDBSTARTUP', N'SLEEP_SYSTEMTASK', N'SLEEP_TASK',
    N'SLEEP_TEMPDBSTARTUP', N'SNI_HTTP_ACCEPT', N'SP_SERVER_DIAGNOSTICS_SLEEP',
    N'SQLTRACE_BUFFER_FLUSH', N'SQLTRACE_INCREMENTAL_FLUSH_SLEEP', N'SQLTRACE_WAIT_
ENTRIES',
    N'WAIT_FOR_RESULTS', N'WAITFOR', N'WAITFOR_TASKSHUTDOWN', N'WAIT_XTP_HOST_WAIT',
    N'WAIT_XTP_OFFLINE_CKPT_NEW_LOG', N'WAIT_XTP_CKPT_CLOSE', N'XE_DISPATCHER_JOIN',
    N'XE_DISPATCHER_WAIT', N'XE_TIMER_EVENT') OPTION (RECOMPILE);

```

Example output:



The screenshot shows a SQL Server query results window with a status bar at the top indicating 110% CPU usage. The window has tabs for 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: '% Signal (CPU) Waits' and '% Resource Waits'. The first row of data shows 3.86 for signal waits and 96.14 for resource waits.

	% Signal (CPU) Waits	% Resource Waits
1	3.86	96.14

Signal waits that exceed 10-15% are typically a sign of CPU pressure.

- Cumulative wait stats are not as useful on an idle instance that is not under load or performance pressure
- Resource waits are non-CPU-related waits

For information about how to troubleshoot high CPU usage issues in SQL Server, see <https://docs.microsoft.com/en-us/troubleshoot/sql/performance/troubleshoot-high-cpu-usage-issues>.

Query 3: Information about operating system memory size and state

Run the following query against the master database to generate basic information about your operating system memory size and state:

```
SELECT total_physical_memory_kb/1024 AS [Physical Memory (MB)],
       available_physical_memory_kb/1024 AS [Available Memory (MB)],
       total_page_file_kb/1024 AS [Total Page File (MB)],
       available_page_file_kb/1024 AS [Available Page File (MB)],
       system_cache_kb/1024 AS [System Cache (MB)],
       system_memory_state_desc AS [System Memory State]
FROM sys.dm_os_sys_memory WITH (NOLOCK) OPTION (RECOMPILE);
```

Query 4: Input/output statistics by file for the current database

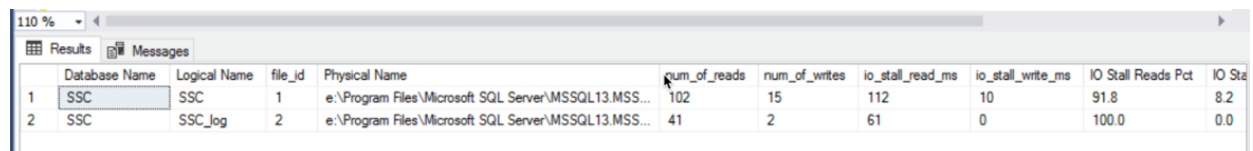
To see input /output (I/O) statistics by file for the current database, run the following query against your Fortify Software Security Center database. This helps you better characterize your workload from an I/O perspective for the Fortify Software Security Center database.

```

SELECT DB_NAME(DB_ID()) AS [Database Name], df.name AS [Logical Name], vfs.[file_id],
df.physical_name AS [Physical Name], vfs.num_of_reads, vfs.num_of_writes, vfs.io_stall_read_ms,
vfs.io_stall_write_ms,
CAST(100. * vfs.io_stall_read_ms/(vfs.io_stall_read_ms + vfs.io_stall_write_ms) AS DECIMAL(10,1))
AS [IO Stall Reads Pct],
CAST(100. * vfs.io_stall_write_ms/(vfs.io_stall_write_ms + vfs.io_stall_read_ms) AS DECIMAL(10,1))
AS [IO Stall Writes Pct],
(vfs.num_of_reads + vfs.num_of_writes) AS [Writes + Reads],
CAST(vfs.num_of_bytes_read/1048576.0 AS DECIMAL(10, 2)) AS [MB Read],
CAST(vfs.num_of_bytes_written/1048576.0 AS DECIMAL(10, 2)) AS [MB Written],
CAST(100. * vfs.num_of_reads/(vfs.num_of_reads + vfs.num_of_writes) AS DECIMAL(10,1)) AS [# Reads
Pct],
CAST(100. * vfs.num_of_writes/(vfs.num_of_reads + vfs.num_of_writes) AS DECIMAL(10,1)) AS [# Write
Pct],
CAST(100. * vfs.num_of_bytes_read/(vfs.num_of_bytes_read + vfs.num_of_bytes_written) AS DECIMAL
(10,1)) AS [Read Bytes Pct],
CAST(100. * vfs.num_of_bytes_written/(vfs.num_of_bytes_read + vfs.num_of_bytes_written) AS DECIMAL
(10,1)) AS [Written Bytes Pct]
FROM sys.dm_io_virtual_file_stats(DB_ID(), NULL) AS vfs
INNER JOIN sys.database_files AS df WITH (NOLOCK)
ON vfs.[file_id]= df.[file_id] OPTION (RECOMPILE);

```

Example output



	Database Name	Logical Name	file_id	Physical Name	num_of_reads	num_of_writes	io_stall_read_ms	io_stall_write_ms	IO Stall Reads Pct	IO Sta
1	SSC	SSC	1	e:\Program Files\Microsoft SQL Server\MSSQL13.MSS...	102	15	112	10	91.8	8.2
2	SSC	SSC_log	2	e:\Program Files\Microsoft SQL Server\MSSQL13.MSS...	41	2	61	0	100.0	0.0

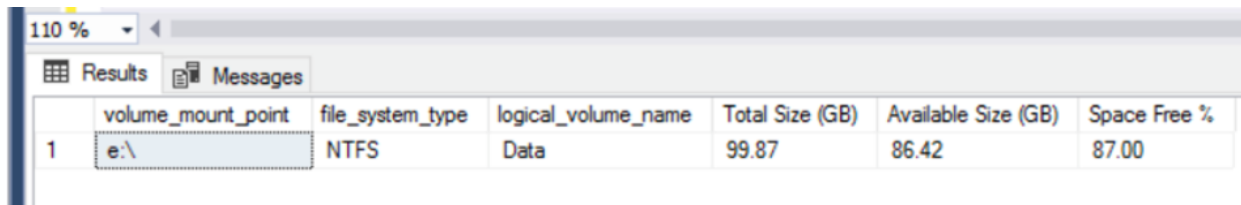
Query 5: Volume information for all logical unit numbers with database files on the current instance

To see details about the logical unit numbers (LUNS) that have database files on the current instance, run the following query against the master database:

```
SELECT DISTINCT vs.volume_mount_point, vs.file_system_type,
vs.logical_volume_name, CONVERT(DECIMAL(18,2),vs.total_bytes/1073741824.0) AS [Total Size (GB)],
CONVERT(DECIMAL(18,2),vs.available_bytes/1073741824.0) AS [Available Size (GB)],
CAST(CAST(vs.available_bytes AS FLOAT)/ CAST(vs.total_bytes AS FLOAT) AS DECIMAL(18,2)) * 100 AS
[Space Free %]
FROM sys.master_files AS f WITH (NOLOCK)
CROSS APPLY sys.dm_os_volume_stats(f.database_id, f.[file_id]) AS vs OPTION (RECOMPILE);
```

This enables you to see the total space and free space on the LUNs where you have database files.

Example output



The screenshot shows a SQL Server Enterprise Manager window with a query results grid. The grid has seven columns: 'volume_mount_point', 'file_system_type', 'logical_volume_name', 'Total Size (GB)', 'Available Size (GB)', and 'Space Free %'. There is one row of data with the following values: 'e:\', 'NTFS', 'Data', '99.87', '86.42', and '87.00'. The 'Results' tab is active, and the window title is '110 %'.

	volume_mount_point	file_system_type	logical_volume_name	Total Size (GB)	Available Size (GB)	Space Free %
1	e:\	NTFS	Data	99.87	86.42	87.00

Query 6: Volume data for all LUNS that have database files on the current instance

To see volume information for all LUNS that have database files on the current instance, run the following query against the master database:

```
CREATE TABLE #IOWarningResults(LogDate datetime, ProcessInfo sysname, LogText nvarchar(1000));
INSERT INTO #IOWarningResults
EXEC xp_readerrorlog 0, 1, N'taking longer than 15 seconds';
INSERT INTO #IOWarningResults
EXEC xp_readerrorlog 1, 1, N'taking longer than 15 seconds';
INSERT INTO #IOWarningResults
EXEC xp_readerrorlog 2, 1, N'taking longer than 15 seconds';
INSERT INTO #IOWarningResults
EXEC xp_readerrorlog 3, 1, N'taking longer than 15 seconds';
INSERT INTO #IOWarningResults
EXEC xp_readerrorlog 4, 1, N'taking longer than 15 seconds';
SELECT LogDate, ProcessInfo, LogText
FROM #IOWarningResults
ORDER BY LogDate DESC;
DROP TABLE #IOWarningResults;
```

Finding 15-second I/O warnings in the SQL Server Error Log is evidence of poor I/O performance (which might have any number of different causes).

Note: No data should be returned from this query, which is a good thing.

Note: Depending on the number of records returned, and the frequency with which they are returned, consult with your storage team to review the errors.

Query 7: Drive-level latency information

To view drive-level latency information, run the following query against the master database:

```

SELECT [Drive],
       CASE
         WHEN num_of_reads = 0 THEN 0
         ELSE (io_stall_read_ms/num_of_reads)
       END AS [Read Latency],
       CASE
         WHEN io_stall_write_ms = 0 THEN 0
         ELSE (io_stall_write_ms/num_of_writes)
       END AS [Write Latency],
       CASE
         WHEN (num_of_reads = 0 AND num_of_writes = 0) THEN 0
         ELSE (io_stall/(num_of_reads + num_of_writes))
       END AS [Overall Latency],
       CASE
         WHEN num_of_reads = 0 THEN 0
         ELSE (num_of_bytes_read/num_of_reads)
       END AS [Avg Bytes/Read],
       CASE
         WHEN io_stall_write_ms = 0 THEN 0
         ELSE (num_of_bytes_written/num_of_writes)
       END AS [Avg Bytes/Write],
       CASE
         WHEN (num_of_reads = 0 AND num_of_writes = 0) THEN 0
         ELSE ((num_of_bytes_read + num_of_bytes_written)/(num_of_reads + num_of_writes))
       END AS [Avg Bytes/Transfer]
FROM (SELECT LEFT(UPPER(mf.physical_name), 2) AS Drive, SUM(num_of_reads) AS num_of_reads,
             SUM(io_stall_read_ms) AS io_stall_read_ms, SUM(num_of_writes) AS num_of_writes,
             SUM(io_stall_write_ms) AS io_stall_write_ms, SUM(num_of_bytes_read) AS num_of_
bytes_read,
             SUM(num_of_bytes_written) AS num_of_bytes_written, SUM(io_stall) AS io_stall
FROM sys.dm_io_virtual_file_stats(NULL, NULL) AS vfs
INNER JOIN sys.master_files AS mf WITH (NOLOCK)
ON vfs.database_id = mf.database_id AND vfs.file_id = mf.file_id
GROUP BY LEFT(UPPER(mf.physical_name), 2)) AS tab
ORDER BY [Overall Latency] OPTION (RECOMPILE);

```

Example output

The following table shows you the drive-level latency for reads and writes, in milliseconds.

	A	B	C	D	E	F	G	H	I
1	Drive	Read Latency	Write Latency	Overall Latency	Avg Bytes/	Avg Bytes/	Avg Bytes/	Transfer	
2	L:	9	1	1	4176300	44932	89435		
3	F:	1	6	4	59790	92844	82874		
4	E:	1	9	7	58306	54905	55540		
5	D:	21	10	15	400171	20037	201246		
6									
7									
8									

Reference table

Milliseconds	Indicative of
< 1	Excellent
< 5	Very good
5 - 10	Good
10 - 20	Marginally acceptable
20 - 100	Bad
100 - 500	Very bad
>500	Extremely bad

Note: Latencies above 20 to 25 ms usually indicate a problem exists with the storage system that hosts your Fortify Software Security Center database. In such cases, contact your storage team to discuss reducing the Read/Write latency.

Query 8: Index Fragmentation

To check for index fragmentation in your Fortify Software Security Center database, run the following query against your database:

```
SELECT OBJECT_NAME(OBJECT_ID), index_id,index_type_desc,index_level,
avg_fragmentation_in_percent,avg_page_space_used_in_percent,page_count
FROM sys.dm_db_index_physical_stats
(DB_ID(N'SSC'), NULL, NULL, NULL , 'SAMPLED')
ORDER BY avg_fragmentation_in_percent DESC
```

Example output

(No column name)	index_id	index_type_desc	index_level	avg_fragmentation_in_percent	avg_page_space_used_in_percent	page_count
1 scan_issue	5	NONCLUSTERED INDEX	0	3.44827586206897	98.4449962935508	29
2 analysisblob	2	NONCLUSTERED INDEX	0	3.44827586206897	97.0318631084754	29
3 ruledescription	2	NONCLUSTERED INDEX	0	3.2258064516129	97.1775265628861	31
4 scan_issue	3	NONCLUSTERED INDEX	0	2.94117647058824	98.8019520632567	34
5 issue	10	NONCLUSTERED INDEX	0	2.63157894736842	79.9058438349395	38
6 issue	4	NONCLUSTERED INDEX	0	2.56410256410256	78.5526562886088	39
8 analysisblob	1	CLUSTERED INDEX	0	1.51515151515152	97.5930565851248	66
9 catpackexternalcategory	1	CLUSTERED INDEX	0	1.14942528735632	97.4611564121572	87
10 ruledescription	1	CLUSTERED INDEX	0	0.813008130081301	99.6523597726711	123
11 scan_issue	1	CLUSTERED INDEX	0	0.675675675675676	93.1183592784779	592
12 issue	1	CLUSTERED INDEX	0	0.38560411311054	87.1366938472943	778
13 catpacklookup	2	NONCLUSTERED INDEX	0	0.269541778975741	99.5610946380035	371
14 catpacklookup	2	NONCLUSTERED INDEX	0	0.269541778975741	99.5610946380035	371

This output indicates the following fragmentation levels in the Fortify Software Security Center database:

avg_fragmentation_in_percent value	Corrective statement
> 5% and <= 30%	ALTER INDEX REORGANIZE
> 30%	ALTER INDEX REBUILD WITH (ONLINE = ON)*

If you have a SQL job configured to run a maintenance plan to rebuild or reorganize the indexes, check out IndexOptimize, SQL Server Maintenance Solution's stored procedure for rebuilding and reorganizing indexes and updating statistics. For details, see SQL Server Maintenance Solution (<https://ola.hallengren.com/sql-server-index-and-statistics-maintenance.html>).

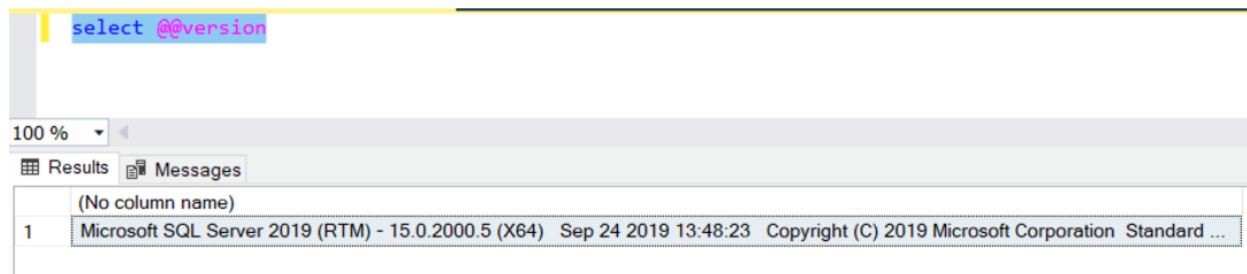
You can also use Microsoft's AdaptiveIndexDefrag to perform an intelligent defrag on one or more indexes, and a required statistics update. For details, see <https://github.com/microsoft/tigertoolbox/tree/master/AdaptiveIndexDefrag>.

Query 9: SQL Version

To determine the SQL version of your database, run the following query:

```
Select @@Version
```

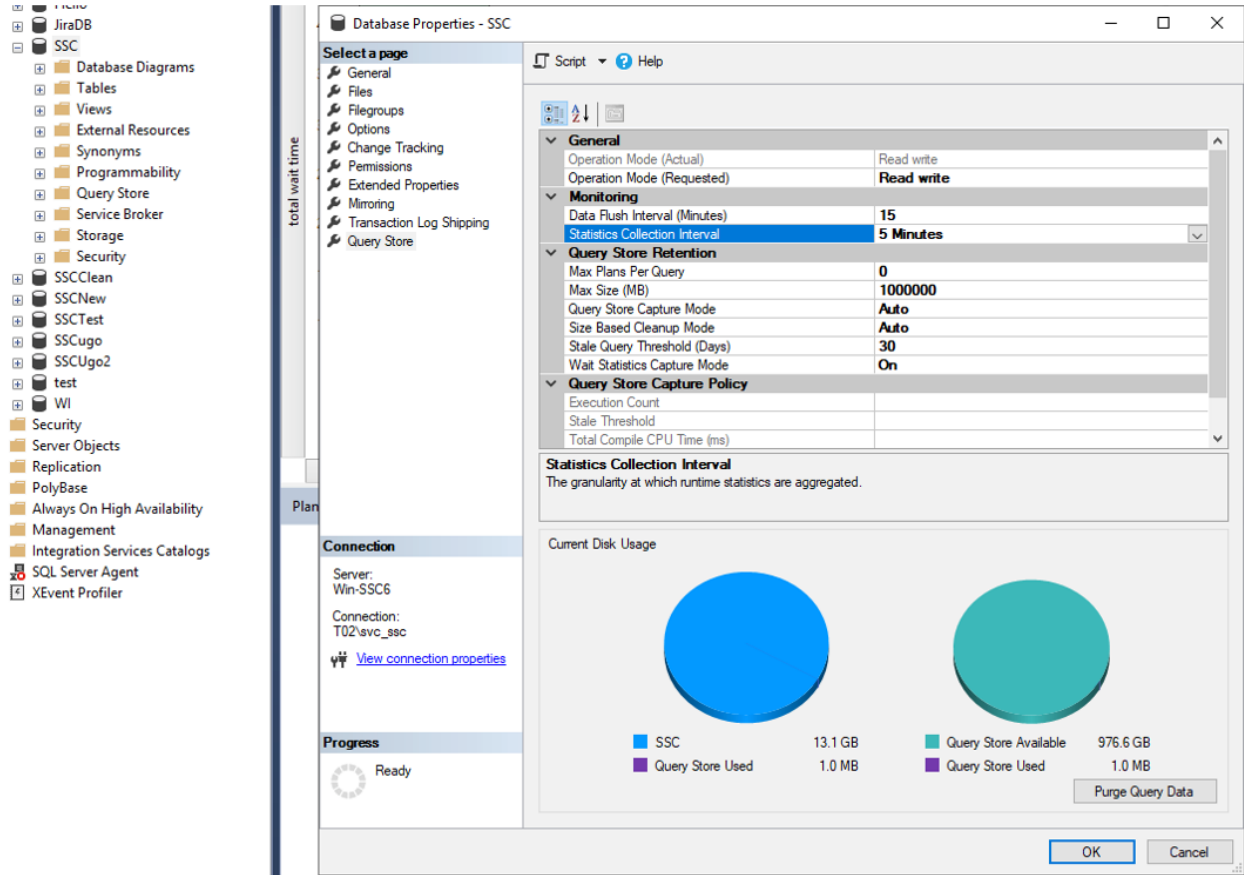
Example output:



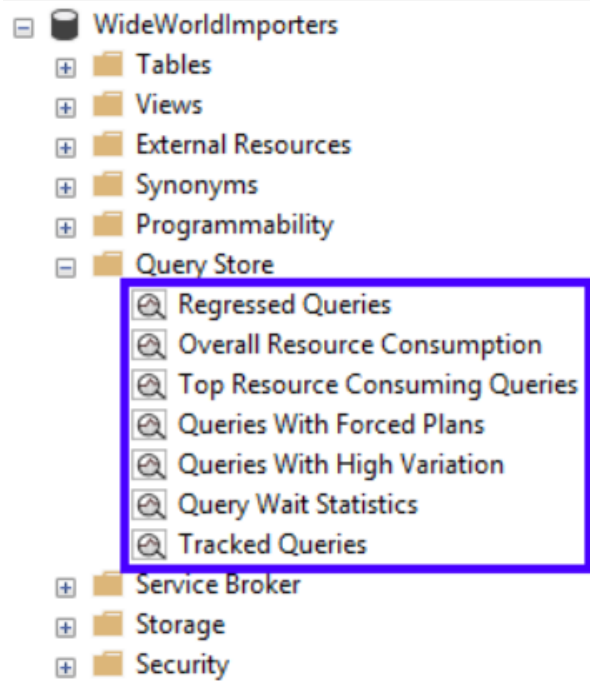
(Optional) Enable Query Store for the Fortify Software Security Center database

Suppose you want to collect data for a minimum of 24 hours. On the SQL Server **Database Properties** page, set the properties with the values shown in the following table:

Property	Setting
Operation Mode	Read Write
Statistics Collection Interval	5 Minutes (You can collect statistics every 1 minute)
Max Size (MB)	The default value is 1000. Add a few zeros to this value. (The data can later be purged.)
Wait Statistics Capture Mode	On



The Query Store includes several queries.



Note: For Azure Synapse Analytics, Query Store views are available under **System Views** in the database portion of the Object Explorer pane.

SQL Scripts: First Responder Kit

Brent Ozar (Microsoft Certified Master, SQL Server Consultant and Trainer) offers a free First Responder Kit (<https://www.brentozar.com/first-aid>) to help you analyze and tune your SQL database. It includes the scripts described in the following table.

Script	Purpose
sp_Blitz	If you acquire a database and you are uncertain about its health, you can run this script to perform a database health assessment that quickly flags common issues. For each issue uncovered, the script provides a link to a web page with more in-depth advice. For details, see the sp_Blitz® Documentation (https://www.brentozar.com/blitz/documentation). To see a video demo on how to use the script, see the "sp_Blitz® – Free SQL Server Health Check Script" web page (https://www.brentozar.com/blitz).
sp_BlitzFirst	This script helps troubleshoot slow SQL Servers by quickly: <ul style="list-style-type: none"> • Blocking long-running queries • Determining whether a backup, database console command (DBCC), or index maintenance job was running • Locating any SQL Server bottlenecks • Checking Perfmon Counters for CPU use, slow drive response times, or low Page Life Expectancy <p>To view a video on how to use the script, see https://www.brentozar.com/askbrent.</p>
sp_BlitzCache	Use this script to determine which queries are causing the biggest performance problems and what you can do about them. For details about the script and to view a video on how to use it, see "sp_BlitzCache®: Find Your Worst-Performing Queries" (https://www.brentozar.com/blitzcache).
sp_BlitzIndex	Use this script to conduct a sanity check and report on your database and diagnose your indexes major disorders. For each detected disorder, a URL is provided that explains what to look for and how to handle the issue. The script also enables you to see both the “missing” and existing indexes for a table in a single view. <p>For more details, and to see a video on how to use the script, see "sp_BlitzIndex® – SQL Server’s Index Sanity Test" (https://www.brentozar.com/blitzindex).</p>

Script	Purpose
sp_BlitzLock	Use this script to analyze deadlocks and determine what queries and tables you need to change. For detailed information about the script, see "Introducing sp_BlitzLock: For Troubleshooting SQL Server Deadlocks" (https://www.brentozar.com/archive/2017/12/introducing-sp_blitzlock-troubleshooting-sql-server-deadlocks).