SQL Server Audit (Database Engine)

11/21/2016 • 12 minutes to read • Contributors

In this article

SQL Server Audit Components
Overview of Using SQL Server Audit
Considerations
Creating and Managing Audits with Transact-SQL
Permissions
Related Tasks
Topics Closely Related to Auditing
See Also

THIS TOPIC APPLIES TO: SQL Server Azure SQL Database (Managed Instance only) Azure SQL Data Warehouse Parallel Data Warehouse

Auditing an instance of the SQL Server Database Engine or an individual database involves tracking and logging events that occur on the Database Engine. SQL Server audit lets you create server audits, which can contain server audit specifications for server level events, and database audit specifications for database level events. Audited events can be written to the event logs or to audit files.

> **Important**
>
> On Azure SQL Database Managed Instance, this T-SQL feature has certain behavior changes. See Azure SQL Database Managed Instance T-SQL differences from SQL Server for details for all T-SQL behavior changes.

There are several levels of auditing for SQL Server, depending on government or standards requirements for your installation. SQL Server Audit provides the tools and processes you must have to enable, store, and view audits on various server and database objects.

You can record server audit action groups per-instance, and either database audit action groups or database audit actions per database. The audit event will occur every time that the auditable action is encountered.

All editions of SQL Server support server level audits. All editions support database level audits beginning with SQL Server 2016 SP1. Prior to that, database level auditing was limited to Enterprise, Developer, and Evaluation editions. For more information, see Features Supported by the Editions of SQL Server 2016.

> **Note**
>
> This topic applies to SQL Server. For SQL Database, see **Get started with SQL database auditing**.

SQL Server Audit Components

An *audit* is the combination of several elements into a single package for a specific group of server actions or database actions. The components of SQL Server audit combine to produce an output that is called an audit, just as a report definition combined with graphics and data elements produces a report.

SQL Server audit uses *Extended Events* to help create an audit. For more information about Extended Events, see Extended Events.

SQL Server Audit

The *SQL Server Audit* object collects a single instance of server or database-level actions and groups of actions to monitor. The audit is at the SQL Server instance level. You can have multiple audits per SQL Server instance.

When you define an audit, you specify the location for the output of the results. This is the audit destination. The audit is created in a *disabled* state, and does not automatically audit any actions. After the audit is enabled, the audit destination receives data from the audit.

Server Audit Specification

The *Server Audit Specification* object belongs to an audit. You can create one server audit specification per audit, because both are created at the SQL Server instance scope.

The server audit specification collects many server-level action groups raised by the Extended Events feature. You can include *audit action groups* in a server audit specification. Audit action groups are predefined groups of actions, which are atomic events occurring in the Database Engine. These actions are sent to the audit, which records them in the target.

Server-level audit action groups are described in the topic SQL Server Audit Action Groups and Actions.

Database Audit Specification

The *Database Audit Specification* object also belongs to a SQL Server audit. You can create one database audit specification per SQL Server database per audit.

The database audit specification collects database-level audit actions raised by the Extended Events feature. You can add either audit action groups or audit events to a database audit specification. *Audit events* are the atomic actions that can be audited by the SQL Server engine. *Audit action groups* are predefined groups of actions. Both are at the SQL Server database scope. These actions are sent to the audit, which records them in the target. Do not include server-scoped objects, such as the system views, in a user database audit specification.

Database-level audit action groups and audit actions are described in the topic SQL Server Audit Action Groups and Actions.

Target

The results of an audit are sent to a target, which can be a file, the Windows Security event log, or the Windows Application event log. Logs must be reviewed and archived periodically to make sure that the target has sufficient space to write additional records.

> **Important**
>
> Any authenticated user can read and write to the Windows Application event log. The Application event log requires lower permissions than the Windows Security event log and is less secure than the Windows Security event log.

Writing to the Windows Security log requires the SQL Server service account to be added to the **Generate security audits** policy. By default, the Local System, Local Service, and Network Service are part of this policy. This setting can be configured by using the security policy snap-in (secpol.msc). Additionally, the **Audit object access** security policy must be enabled for both **Success** and **Failure**. This setting can be configured by using the security policy snap-in (secpol.msc). In Windows Vista or Windows Server 2008, you can set the more granular **application generated** policy from the command line by using the audit policy program (**AuditPol.exe)**. For more information about the steps to enable writing to the Windows Security log, see Write SQL Server Audit Events to the Security Log. For more information about the Auditpol.exe program, see Knowledge Base article 921469, How to use Group Policy to configure detailed security auditing. The Windows event logs are global to the Windows operating system. For more information about the Windows event logs, see Event Viewer Overview. If you need more precise permissions on the audit, use the binary file target.

When you are saving audit information to a file, to help prevent tampering, you can restrict access to the file location in the following ways:

- The SQL Server Service Account must have both Read and Write permission.

- Audit Administrators typically require Read and Write permission. This assumes that the Audit Administrators are Windows accounts for administration of audit files, such as: copying them to different shares, backing them up, and so on.
- Audit Readers that are authorized to read audit files must have Read permission.

  Even when the Database Engine is writing to a file, other Windows users can read the audit file if they have permission. The Database Engine does not take an exclusive lock that prevents read operations.

  Because the Database Engine can access the file, SQL Server logins that have CONTROL SERVER permission can use the Database Engine to access the audit files. To record any user that is reading the audit file, define an audit on master.sys.fn_get_audit_file. This records the logins with CONTROL SERVER permission that have accessed the audit file through SQL Server.

  If an Audit Administrator copies the file to a different location (for archive purposes, and so on), the ACLs on the new location should be reduced to the following permissions:

- Audit Administrator – Read / Write
- Audit Reader – Read

  We recommend that you generate audit reports from a separate instance of SQL Server, such as an instance of SQL Server Express, to which only Audit Administrators or Audit Readers have access. By using a separate instance of the Database Engine for reporting, you can help prevent unauthorized users from obtaining access to the audit record.

  You can offer additional protection against unauthorized access by encrypting the folder in which the audit file is stored by using Windows BitLocker Drive Encryption or Windows Encrypting File System.

  For more information about the audit records that are written to the target, see SQL Server Audit Records.

Overview of Using SQL Server Audit

You can use SQL Server Management Studio or Transact-SQL to define an audit. After the audit is created and enabled, the target will receive entries.

You can read the Windows event logs by using the **Event Viewer** utility in Windows. For file targets, you can use either the **Log File Viewer** in SQL Server Management Studio or the fn_get_audit_file function to read the target file.

The general process for creating and using an audit is as follows.

1. Create an audit and define the target.

2. Create either a server audit specification or database audit specification that maps to the audit. Enable the audit specification.

3. Enable the audit.

4. Read the audit events by using the Windows **Event Viewer**, **Log File Viewe**r, or the fn_get_audit_file function.

   For more information, see Create a Server Audit and Server Audit Specification and Create a Server Audit and Database Audit Specification.

Considerations

In the case of a failure during audit initiation, the server will not start. In this case, the server can be started by using the **–f** option at the command line.

When an audit failure causes the server to shut down or not to start because ON_FAILURE=SHUTDOWN is specified for the audit, the MSG_AUDIT_FORCED_SHUTDOWN event will be written to the log. Because the shutdown will occur on the first encounter of this setting, the event will be written one time. This event is written after the failure message for the audit causing the shutdown. An administrator can bypass audit-induced shutdowns by starting SQL Server in Single User mode using the **–m** flag. If you start in Single User mode, you will downgrade any audit where ON_FAILURE=SHUTDOWN is specified to run in that session as ON_FAILURE=CONTINUE. When SQL Server is started by using the **–m** flag, the MSG_AUDIT_SHUTDOWN_BYPASSED message will be written to the error log.

For more information about service startup options, see Database Engine Service Startup Options.

Attaching a Database with an Audit Defined

Attaching a database that has an audit specification and specifies a GUID that does not exist on the server will cause an *orphaned* audit specification. Because an audit with a matching GUID does not exist on the server instance, no audit events will be recorded. To correct this situation, use the ALTER DATABASE AUDIT SPECIFICATION command to connect the orphaned audit specification to an existing server audit. Or, use the CREATE SERVER AUDIT command to create a new server audit with the specified GUID.

You can attach a database that has an audit specification defined on it to another edition of SQL Server that does not support SQL Server audit, such as SQL Server Express but it will not record audit events.

Database Mirroring and SQL Server Audit

A database that has a database audit specification defined and that uses database mirroring will include the database audit specification. To work correctly on the mirrored SQL instance, the following items must be configured:

- The mirror server must have an audit with the same GUID to enable the database audit specification to write audit records. This can be configured by using the command CREATE AUDIT WITH GUID= *<GUID from source Server Audit>*.
- For binary file targets, the mirror server service account must have appropriate permissions to the location where the audit trail is being written.
- For Windows event log targets, the security policy on the computer where the mirror server is located must allow for service account access to the security or application event log.

Auditing Administrators

Members of the **sysadmin** fixed server role are identified as the **dbo** user in each database. To audit actions of the administrators, audit the actions of the **dbo** user.

Creating and Managing Audits with Transact-SQL

You can use DDL statements, dynamic management views and functions, and catalog views to implement all aspects of SQL Server Audit.

Data Definition Language Statements

You can use the following DDL statements to create, alter, and drop audit specifications:

ALTER AUTHORIZATION	CREATE SERVER AUDIT
ALTER DATABASE AUDIT SPECIFICATION	CREATE SERVER AUDIT SPECIFICATION
ALTER SERVER AUDIT	DROP DATABASE AUDIT SPECIFICATION
ALTER SERVER AUDIT SPECIFICATION	DROP SERVER AUDIT
CREATE DATABASE AUDIT SPECIFICATION	DROP SERVER AUDIT SPECIFICATION

Dynamic Views and Functions

The following table lists the dynamic views and function that you can use for SQL Server Auditing.

Dynamic views and functions	Description
sys.dm_audit_actions	Returns a row for every audit action that can be reported in the audit log and every audit action group that can be configured as part of SQL Server Audit.

sys.dm_server_audit_status	Provides information about the current state of the audit.
sys.dm_audit_class_type_map	Returns a table that maps the class_type field in the audit log to the class_desc field in sys.dm_audit_actions.
fn_get_audit_file	Returns information from an audit file created by a server audit.

Catalog Views

The following table lists the catalog views that you can use for SQL Server auditing.

Catalog views	Description
sys.database_ audit_specifications	Contains information about the database audit specifications in a SQL Server audit on a server instance.
sys.database_audit_specification_details	Contains information about the database audit specifications in a SQL Server audit on a server instance for all databases.
sys.server_audits	Contains one row for each SQL Server audit in a server instance.
sys.server_audit_specifications	Contains information about the server audit specifications in a SQL Server audit on a server instance.
sys.server_audit_specifications_details	Contains information about the server audit specification details (actions) in a SQL Server audit on a server instance.
sys.server_file_audits	Contains stores extended information about the file audit type in a SQL Server audit on a server instance.

Permissions

Each feature and command for SQL Server Audit has individual permission requirements.

To create, alter, or drop a Server Audit or Server Audit Specification, server principals require the ALTER ANY SERVER AUDIT or the CONTROL SERVER permission. To create, alter, or drop a Database Audit Specification, database principals require the ALTER ANY DATABASE AUDIT permission or the ALTER or CONTROL permission on the database. In addition, principals must have permission to connect to the database, or ALTER ANY SERVER AUDIT or CONTROL SERVER permissions.