sys.fn_get_audit_file (Transact-SQL)

05/16/2017 • 7 minutes to read • Contributors all

In this article

Syntax
Arguments
Tables Returned
Remarks
Permissions
Examples
See Also

THIS TOPIC APPLIES TO: SQL Server (starting with 2008) Azure SQL Database Azure SQL Data Warehouse Parallel Data Warehouse

Returns information from an audit file created by a server audit in SQL Server. For more information, see SQL Server Audit (Database Engine).

Transact-SQL Syntax Conventions

Syntax

Copy

```
fn_get_audit_file ( file_pattern,
    { default | initial_file_name | NULL },
    { default | audit_record_offset | NULL } )
```

Arguments

file_pattern

Specifies the directory or path and file name for the audit file set to be read. Type is **nvarchar(26**

FEEDBACK

- **SQL Server**:

  This argument must include both a path (drive letter or network share) and a file name that can include a wildcard. A single asterisk (*) can be used to collect multiple files from an audit file set. For example:

  - **<path>*** - Collect all audit files in the specified location.
  - **<path>\LoginsAudit_{GUID}** - Collect all audit files that have the specified name and GUID pair.
  - **<path>\LoginsAudit_{GUID}_00_29384.sqlaudit** - Collect a specific audit file.

- **Azure SQL Database**:

  This argument is used to specify a blob URL (including the storage endpoint and container). While it does not support an asterisk wildcard, you can use a partial file (blob) name prefix (instead of the full blob name) to collect multiple files (blobs) that begin with this prefix. For example:

  - **<Storage_endpoint>/<Container>/<ServerName>/<DatabaseName>/** - collects all audit files (blobs) for the specific database.
  - **<Storage_endpoint>/<Container>/<ServerName>/<DatabaseName>/<AuditName>/<Crea** - collects a specific audit file (blob).

> ⓘ **Note**
>
> Passing a path without a file name pattern will generate an error.

initial_file_name
Specifies the path and name of a specific file in the audit file set to start reading audit records from. Type is **nvarchar(260)**.

> ⓘ **Note**
>
> The *initial_file_name* argument must contain valid entries or must contain either the default | NULL value.

audit_record_offset
Specifies a known location with the file specified for the initial_file_name. When this argument is used the function will start reading at the first record of the Buffer immediately following the specified offset.

> ⓘ **Note**

The *audit_record_offset* argument must contain valid entries or must contain either the default | NULL value. Type is **bigint**.

Tables Returned

The following table describes the audit file content that can be returned by this function.

Column name	Type	Description
event_time	**datetime2**	Date and time when the auditable action is fired. Is not nullable.
sequence_number	**int**	Tracks the sequence of records within a single audit record that was too large to fit in the write buffer for audits. Is not nullable.
action_id	**varchar(4)**	ID of the action. Is not nullable.
succeeded	**bit**	Indicates whether the action that triggered the event succeeded. Is not nullable. For all events other than login events, this only reports whether the permission check succeeded or failed, not the operation. 1 = success 0 = fail
permission_bitmask	**varbinary(16)**	In some actions, this is the permissions that were grant, denied, or revoked.
is_column_permission	**bit**	Flag indicating if this is a column level permission. Is not nullable. Returns 0 when the permission_bitmask = 0. 1 = true 0 = false
session_id	**smallint**	ID of the session on which the event occurred. Is not nullable.
server_principal_id	**int**	ID of the login context that the action is performed in. Is not nullable.

database_principal_id	**int**	ID of the database user context that the action is performed in. Is not nullable. Returns 0 if this does not apply. For example, a server operation.
target_server_principal_id	**int**	Server principal that the GRANT/DENY/REVOKE operation is performed on. Is not nullable. Returns 0 if not applicable.
target_database_principal_id	**int**	The database principal the GRANT/DENY/REVOKE operation is performed on. Is not nullable. Returns 0 if not applicable.
object_id	**int**	The ID of the entity on which the audit occurred. This includes the following: Server objects Databases Database objects Schema objects Is not nullable. Returns 0 if the entity is the Server itself or if the audit is not performed at an object level. For example, Authentication.
class_type	**varchar(2)**	The type of auditable entity that the audit occurs on. Is not nullable.
session_server_principal_name	**sysname**	Server principal for session. Is nullable.
server_principal_name	**sysname**	Current login. Is nullable.
server_principal_sid	**varbinary**	Current login SID. Is nullable.
database_principal_name	**sysname**	Current user. Is nullable. Returns NULL if not available.
target_server_principal_name	**sysname**	Target login of action. Is nullable. Returns NULL if not applicable.
target_server_principal_sid	**varbinary**	SID of target login. Is nullable. Returns NULL if not applicable.
target_database_principal_name	**sysname**	Target user of action. Is nullable.

		Returns NULL if not applicable.
server_instance_name	**sysname**	Name of the server instance where the audit occurred. The standard server\instance format is used.
database_name	**sysname**	The database context in which the action occurred. Is nullable. Returns NULL for audits occuring at the server level.
schema_name	**sysname**	The schema context in which the action occurred. Is nullable. Returns NULL for audits occuring outside a schema.
object_name	**sysname**	The name of the entity on which the audit occurred. This includes the following: Server objects Databases Database objects Schema objects Is nullable. Returns NULL if the entity is the Server itself or if the audit is not performed at an object level. For example, Authentication.
statement	**nvarchar(4000)**	TSQL statement if it exists. Is nullable. Returns NULL if not applicable.
additional_information	**nvarchar(4000)**	Unique information that only applies to a single event is returned as XML. A small number of auditable actions contain this kind of information. One level of TSQL stack will be displayed in XML format for actions that have TSQL stack associated with them. The XML format will be: `<tsql_stack><frame nest_level = '%u' database_name = '%.*s' schema_name = '%.*s' object_name = '%.*s' /> </tsql_stack>`

		Frame nest_level indicates the current nesting level of the frame. The Module name is represented in three part format (database_name, schema_name and object_name). The module name will be parsed to escape invalid xml characters like `'\<'`, `'>'`, `'/'`, `'_x'`. They will be escaped as `_xHHHH\_`. The HHHH stands for the four-digit hexadecimal UCS-2 code for the character Is nullable. Returns NULL when there is no additional information reported by the event.
file_name	**varchar(260)**	The path and name of the audit log file that the record came from. Is not nullable.
audit_file_offset	**bigint**	The buffer offset in the file that contains the audit record. Is not nullable.
user_defined_event_id	**smallint**	**Applies to**: SQL Server 2012 through SQL Server 2017. User defined event id passed as an argument to **sp_audit_write**. **NULL** for system events (default) and non-zero for user-defined event. For more information, see sp_audit_write (Transact-SQL).
user_defined_information	**nvarchar(4000)**	**Applies to**: SQL Server 2012 through SQL Server 2017. Used to record any extra information the user wants to record in
audit_schema_version	**int**	
sequence_group_id	**varbinary**	**Applies to**: SQL Server only (starting with 2016)

transaction_id	**bigint**	**Applies to**: SQL Server only (starting with 2016)
client_ip	**nvarchar(128)**	**Applies to**: Azure SQL DB + SQL Server (starting with 2017)
application_name	**nvarchar(128)**	**Applies to**: Azure SQL DB + SQL Server (starting with 2017)
duration_milliseconds	**bigint**	**Applies to**: Azure SQL DB only
response_rows	**bigint**	**Applies to**: Azure SQL DB only
affected_rows	**bigint**	**Applies to**: Azure SQL DB only
connection_id	GUID	**Applies to**: Azure SQL DB only
data_sensitivity_information	nvarchar(4000)	**Applies to**: Azure SQL DB only

Remarks

If the *file_pattern* argument passed to **fn_get_audit_file** references a path or file that does not exist, or if the file is not an audit file, the **MSG_INVALID_AUDIT_FILE** error message is returned.

Permissions

- **SQL Server**: Requires the **CONTROL SERVER** permission.
- **Azure SQL DB**: Requires the **CONTROL DATABASE** permission.
    - Server admins can access audit logs of all databases on the server.
    - Non server admins can only access audit logs from the current database.
    - Blobs that do not meet the above criteria will be skipped (a list of skipped blobs will be displayed in the query output message), and the function will return logs only from blobs for which access is allowed.

Examples

- **SQL Server**

  This example reads from a file that is named `\\serverName\Audit\HIPPA_AUDIT.sqlaudit`.

  Copy