Splunk® Supported Add-ons
Splunk Add-on for Microsoft SQL Server released

Generated: 6/26/2022 1:04 am

Table of Contents

Overview

The Splunk Add-on for Microsoft SQL Server

Version	3.0.0
Vendor products	Microsoft SQL Server 2012 Enterprise, Microsoft SQL Server 2014 Enterprise, Microsoft SQL Server 2016 Enterprise, Microsoft SQL Server 2019 Enterprise, Microsoft SQL Server 2017 Standard
Add-on has a web UI	No. This add-on does not contain any views.

The Splunk Add-on for Microsoft SQL Server allows a Splunk software administrator to collect system performance, SQL server performance, log, audit, and status data from Microsoft SQL Server deployments.

The Splunk Add-on for Microsoft SQL Server uses Splunk DB Connect, Splunk Windows Performance monitoring, and file monitoring to collect data. Through log file monitoring and field extraction, the database administrator can correlate events and create alerts and dashboards to track database errors, problems, or incidents in real time.

This add-on provides the inputs and **CIM**-compatible knowledge to use with other Splunk apps, such as Splunk Enterprise Security, the Splunk App for PCI Compliance, and Splunk IT Service Intelligence.

Download the Splunk Add-on for Microsoft SQL Server from Splunkbase.

For a summary of new features, fixed issues, and known issues, see Release Notes for the Splunk Add-on for Microsoft SQL Server.

For information about installing and configuring the Splunk Add-on for Microsoft SQL Server, see Installation and configuration overview for the Splunk Add-on for Microsoft SQL Server.

Search the Splunk Community page for more information about this add-on.

Source types for the Splunk Add-on for Microsoft SQL Server

The Splunk Add-on for Microsoft SQL Server collects different kinds of data from Microsoft SQL Server and assigns a source type for each kind of data. It collects data via file monitoring, Windows Performance Monitoring, and through Splunk DB Connect:

Source types collected through file monitoring

Log	Log Format	Description	Source Type	File Location	CIM data models
Error log	Plain text	The error log contains error messages as well as some activities of SQL Server.	`mssql:errorlog`	After you install and start Microsoft SQL Server, the server creates this log file under the SQL Server installation folder. Example Location: `C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Log\ERRORLOG*`	Authentication
	Plain text		`mssql:agentlog`		None

Log	Log Format	Description	Source Type	File Location	CIM data models
Agent log		The agent log records SQL Server agent service related activities.		After you install and start the Microsoft SQL Server agent, the server creates this log file under the SQL Server installation folder. Example Location: `C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Log\SQLAGENT.OUT`	-

Source types collected through Windows Performance Monitoring

Object	Counter	Source type	CIM data models
Processor	% Processor Time	`perfmon:sqlserverhost:processor`	None
Network Interface	Current Bandwidth; Bytes Total/sec	`perfmon:sqlserverhost:network`	None
Memory	% Committed Bytes In Use; Pages/sec; Available Mbytes; Pages	`perfmon:sqlserverhost:memory`	None
SQLServer:Buffer Manager	*	`perfmon:sqlserver:buffer_manager`	None
SQLServer:Databases	Active Transactions; Data File(s) Size (KB); Log File(s) Size (KB);Log File(s) Used Size (KB); Transactions/sec	`perfmon:sqlserver:databases`	None
SQLServer:Memory Manager	Total Server Memory (KB);Granted Workspace Memory (KB);Maximum Workspace Memory (KB);Memory Grants Outstanding;Memory Grants Pending;Target Server Memory (KB)	`perfmon:sqlserver:memory_manager`	None
LogicalDisk	Avg. Disk sec/Read; Avg. Disk sec/Write	`perfmon:sqlserverhost:logicaldisk`	None
PhysicalDisk	Disk Reads/sec; Disk Writes/sec; Avg. Disk sec/Read; Avg. Disk sec/Write; Avg. Disk sec/Transfer; Disk Read Bytes/sec; Disk Write Bytes/sec; Avg. Disk Queue Length	`perfmon:sqlserverhost:physicaldisk`	None
Paging File	% Usage; % Usage Peak	`perfmon:sqlserverhost:paging_file`	None
Process	Private Bytes; % Processor Time	`perfmon:sqlserverhost:process`	None
System	Processor Queue Length; Context Switches/sec	`perfmon:sqlserverhost:system`	None
SQLServer:General Statistics	User Connections; Processes blocked; Logins/sec; Logout/sec	`perfmon:sqlserver:general_statistics`	None
SQLServer:SQL Statistics	Batch Requests/sec; SQL Compilations/sec; SQL re-Compilations/sec; SQL Attention Rate/sec; Auto-Param Attempts/sec; Failed Auto-Params/sec; Safe Auto-Params/sec; Unsafe Auto-Params/sec	`perfmon:sqlserver:sql_statistics`	None
SQLServer:Access Methods	Forwarded Records/sec; Full Scans/sec; Index Searches/sec;Page Splits/sec; Workfiles Created/sec; Worktables Created/sec; Worktables From Cache Ratio; Table Lock Escalations/sec	`perfmon:sqlserver:access_methods`	None
SQLServer:Latches	Latch Waits/sec; Avg Latch Wait Time (ms); Total Latch Wait Time (ms)	`perfmon:sqlserver:latches`	None
SQLServer:SQL Errors	Errors/sec	`perfmon:sqlserver:sql_errors`	None

Object	Counter	Source type	CIM data models
SQLServer:Locks	Number of Deadlocks/sec; Average Wait Time (ms)	`perfmon:sqlserver:locks`	None
SQLServer:Transactions	Transactions; Longest Transaction Running Time	`perfmon:sqlserver:transactions`	None

Source types collected through Splunk DB Connect

Data from Dynamic Management View

Type	Dynamic Management View	Source type	CIM or ITSI data models
alwayson	`sys.dm_hadr_auto_` page_repair	`mssql:alwayson:dm_hadr_auto_page_repair`	None
	`sys.dm_hadr_` availability_group_states	`mssql:alwayson:` dm_hadr_availability_group_states	None
	`sys.dm_hadr_` availability_replica_ cluster_nodes	`mssql:alwayson:` dm_hadr_availability_replica_cluster_nodes	None
	`sys.dm_hadr_` availability_replica_ cluster_states	`mssql:alwayson:` dm_hadr_availability_replica_cluster_states	None
	`sys.dm_hadr_` availability_replica_states	`mssql:alwayson:` dm_hadr_availability_replica_states	None
	`sys.dm_hadr_cluster`	`mssql:alwayson:dm_hadr_cluster`	None
	`sys.dm_hadr_cluster_` members	`mssql:alwayson:dm_hadr_cluster_members`	None
	`sys.dm_hadr_cluster_` networks	`mssql:alwayson:dm_hadr_cluster_networks`	None
	`sys.dm_hadr_database_` replica_cluster_states	`mssql:alwayson:` dm_hadr_database_replica_cluster_states	None
	`sys.dm_hadr_database_` replica_states	`mssql:alwayson:dm_hadr_database_replica_states`	None
	`sys.dm_hadr_instance_` node_map	`mssql:alwayson:dm_hadr_instance_node_map`	None

	Dynamic Management View	Source type	CIM or ITSI data models
	sys.dm_hadr_name_ id_map	mssql:alwayson:dm_hadr_name_id_map	None
database **Type**	sys.dm_tcp_listener_ states	mssql:alwayson:dm_tcp_listener_states	None
	sys.dm_db_file_ space_usage	mssql:database:dm_db_file_space_usage	None
	sys.dm_db_ partition_stats	mssql:database:dm_db_partition_stats	None
	sys.dm_db_ session_space_usage	mssql:database:dm_db_session_space_usage	None
	sys.dm_db_ uncontained_entities	mssql:database:dm_db_uncontained_entities	None
	sys.dm_db_fts_ index_physical_stats	mssql:database:dm_db_fts_index_physical_stats	None
	sys.dm_db_ persisted_sku_features	mssql:database:dm_db_persisted_sku_features	None
	sys.dm_db_ task_space_usage	mssql:database:dm_db_task_space_usage	None
execution	sys.dm_exec_ query_stats	mssql:execution:dm_exec_query_stats	Databases, Database (ITSI)
	sys.dm_exec_ sessions	mssql:execution:dm_exec_sessions	Databases, Database (ITSI)
	sys.dm_exec_ background_job_queue	mssql:execution:dm_exec_background_job_queue	None
	sys.dm_exec_ background_job_queue_stats	mssql:execution: dm_exec_background_job_queue_stats	None
	sys.dm_exec_ cached_plans	mssql:execution:dm_exec_cached_plans	None

Type	Dynamic Management View	Source type	CIM or ITSI data models
	sys.dm_exec_ connections	mssql:execution:dm_exec_connections	None
	sys.dm_exec_ procedure_stats	mssql:execution:dm_exec_procedure_stats	None
	sys.dm_exec_ query_memory_grants	mssql:execution:dm_exec_query_memory_grants	None
	sys.dm_exec_ query_optimizer_info	mssql:execution:dm_exec_query_optimizer_info	None
	sys.dm_exec_ query_resource_semaphores	mssql:execution: dm_exec_query_resource_semaphores	None
	sys.dm_exec_ requests	mssql:execution:dm_exec_requests	None
	sys.dm_exec_ trigger_stats	mssql:execution:dm_exec_trigger_stats	None
index	sys.dm_db_ index_physical_stats	mssql:index:dm_db_index_physical_stats	None
	sys.dm_db_ index_operational_stats	mssql:index:dm_db_index_operational_stats	None
	sys.dm_db_ index_usage_stats	mssql:index:dm_db_index_usage_stats	None
	sys.dm_db_ missing_index_details	mssql:index:dm_db_missing_index_details	None
	sys.dm_db_ missing_index_groups	mssql:index:dm_db_missing_index_groups	None
	sys.dm_db_ missing_index_group_stats	mssql:index: dm_db_missing_index_group_stats	None
instance	Built-in functions: • SERVERPROPERTY • @@MAX_CONNECTIONS	mssql:instance	Databases, Database (ITSI)

Type	Dynamic Management View	Source type	CIM or ITSI data models
	• db_name()		
	Built-in functions: • SERVERPROPERTY • db_name() • @@TOTAL_READ • @@TOTAL_WRITE • @@TOTAL_ERRORS	mssql:instancestats	Databases
	sys.processes	mssql:processes	None
	sys.databases	mssql:databases	Databases
mirroring	sys.dm_db_ mirroring_connections	mssql:mirroring: dm_db_mirroring_connections	None
	sys.dm_db_ mirroring_auto_page_repair	mssql:mirroring: dm_db_mirroring_auto_page_repair	None
OS	sys.dm_os_ sys_info	mssql:os:dm_os_sys_info	Databases; Performance
	sys.dm_os_ performance_counters	mssql:os:dm_os_performance_counters	Database (ITSI)
	sys.dm_os_ windows_info	mssql:os:dm_os_windows_info	None
	sys.dm_os_ buffer_descriptors	mssql:os:dm_os_buffer_descriptors	None
replication	sys.dm_repl_ articles	mssql:replication:dm_repl_articles	None
	sys.dm_repl_ tranhash	mssql:replication:dm_repl_tranhash	None
	sys.dm_repl_ schemas	mssql:replication:dm_repl_schemas	None
	sys.dm_repl_ traninfo	mssql:replication:dm_repl_traninfo	None
transaction	sys.dm_tran_ locks	mssql:transaction:dm_tran_locks	Databases

	Dynamic Management View	Source type	CIM or ITSI data models
Type	sys.dm_tran_active_ snapshot_database_ transactions	mssql:transaction: dm_tran_active_snapshot_database_transactions	None
	sys.dm_tran_current_ snapshot	mssql:transaction:dm_tran_current_snapshot	None
	sys.dm_tran_database_ transactions	mssql:transaction: dm_tran_database_transactions	None
	sys.dm_tran_session_ transactions	mssql:transaction:dm_tran_session_transactions	None
	sys.dm_tran_ transactions_snapshot	mssql:transaction: dm_tran_transactions_snapshot	None
	sys.dm_tran_ active_transactions	mssql:transaction:dm_tran_active_transactions	None
	sys.dm_tran_ current_transaction	mssql:transaction:dm_tran_current_transaction	None
	sys.dm_tran_ top_version_generators	mssql:transaction: dm_tran_top_version_generators	None
	sys.dm_tran_version_store	mssql:transaction:dm_tran_version_store	None
Other	sys.tables	mssql:table	Database (ITSI)
	sys.database_principals	mssql:user	Database (ITSI)

Trace logs

Log	Log Format	Description	Source type	CIM data models
Trace log	Binary	Default trace provides troubleshooting support. You can open default trace logs with SQL Server Profiler or query them with Transact-SQL by using the fn_trace_gettable system function. This add-on uses the fn_trace_gettable system function via DB Connect.	mssql:trclog	None

Audit logs

Log	Log Format	Description	Source Type	Event Type	CIM data models

Audit log	Binary	SQL Server audit lets you create server audits for server-level, database-level, and table-level events. See Create audit objects in Microsoft SQL Server for more information. Audit logs can be read by the sys.fn_get_audit_file system function. This add-on uses the sys.fn_get_audit_file function via DB Connect.	mssql:audit	microsoft_sqlserver_audit_logout	Change
				microsoft_sqlserver_audit_reset_password	Change
				microsoft_sqlserver_audit_logout	Change
				microsoft_sqlserver_audit_trace	Change
				microsoft_sqlserver_audit_login	Authentication

Release notes for the Splunk Add-on for Microsoft SQL Server

Version 3.0.0 of the Splunk Add-on for Microsoft SQL Server was released on December 8, 2021.

3.0.0

Version 3.0.0 of the Splunk Add-on for Microsoft SQL Server is compatible with the following software, CIM versions, and platforms.

Splunk platform versions	8.1.x and 8.2.x
Splunk DB Connect	3.6.0 and later
CIM	4.20
Platforms	Windows for local data collection on MS SQL Server, platform independent otherwise
Vendor Products	Microsoft SQL Server 2012 Enterprise, Microsoft SQL Server 2014 Enterprise, Microsoft SQL Server 2016 Enterprise, Microsoft SQL Server 2017 Enterprise, Microsoft SQL Server 2017 Standard, Microsoft SQL Server 2019 Enterprise, Microsoft SQL Server 2019 Standard

The field alias functionality is compatible with the current version of this add-on. The current version of this add-on does not support older field alias configurations.

For more information about the field alias configuration change, refer to the Splunk Enterprise Release Notes.

New Features

- Support for Microsoft SQL Server Standard 2019.
- Drop support for Splunk DB 2.x.x.
- Compatibility with Splunk DB 3.6.0 and later.

Common Information Model (CIM) enhancements:

- Added new field extractions for the `mssql:audit` sourcetype.
- Support for version 4.20.

For information on upgrading to the newest version of this add-on, see the Upgrade the Splunk Add-on for Microsoft SQL Server topic in this manual.

Known issues

Version 3.0.0 of the Splunk Add-on for Microsoft SQL Server has the following known issues.

If no issues appear below, no issues have yet been reported:

Third-party software attributions

Version 3.0.0 of the Splunk Add-on for Microsoft SQL Server does not incorporate any third-party components or libraries.

Release history for the Splunk Add-on for Microsoft SQL Server

Latest release

The latest version of the Splunk Add-on for Microsoft SQL Server is version 3.0.0. See Release notes for the Splunk Add-on for Microsoft SQL Server for the release notes of this latest version.

Version 2.0.0

Version 2.0.0 of the Splunk Add-on for Microsoft SQL Server is compatible with the following software, CIM versions, and platforms.

Splunk platform versions	7.2.x, 7.3.x, 8.0.x and 8.1.0
Splunk DB Connect	2.4.1, 3.1.3, 3.3.1 and 3.4.0
CIM	4.17
Platforms	Windows for local data collection on MS SQL Server, platform independent otherwise
Vendor products	Microsoft SQL Server 2012 Enterprise, Microsoft SQL Server 2014 Enterprise, Microsoft SQL Server 2016 Enterprise, Microsoft SQL Server 2017 Enterprise, Microsoft SQL Server 2017 Standard, Microsoft SQL Server 2019 Enterprise, Microsoft SQL Server 2019 Standard.

The field alias functionality is compatible with the current version of this add-on. The current version of this add-on does not support older field alias configurations.

For more information about the field alias configuration change, refer to the Splunk Enterprise Release Notes.

New Features

- Support for Microsoft SQL Server Standard 2017 and Microsoft SQL Server Enterprise 2019.
- Compatibility with Splunk DB Connect 3.3.1 and 3.4.0.
- Added field extractions for the `mssql:errorlog` and `mssql:agentlog` sourcetypes.
- Removed the search time extractions of the `host` and `port` field.
    - The value of the host will be the same as the host provided at the time of connection in Splunk DB Connect.

    - For the port field, updated the SQL queries so it will be populated at index-time in the event.
- Common Information Model (CIM) enhancements:
    - Support for version 4.17.
    - Authentication data model mapping for the logon events in the `mssql:errorlog` sourcetype.
    - Databases data model mapping for the `mssql:databases` sourcetype.
    - Removed the `serial_num` field from the `mssql:transaction:dm_tran_locks` sourcetype.
    - Additional Splunk IT Service Intelligence (ITSI) database module field compatibility.

For information on upgrading to the newest version of this add-on, see the Upgrade the Splunk Add-on for Microsoft SQL Server topic in this manual.

Fixed issues

Version 2.0.0 of the Splunk Add-on for Microsoft SQL Server has the following fixed issues.

Date resolved	Issue number	Description
2020-10-22	ADDON-30436	'Additional_Information' field is not getting extracted properly for the 'mssql:audit' sourcetype
2020-09-21	ADDON-29421	Removed incorrect field mapping of serial_num in sourcetype = mssql:transaction:dm_tran_locks

Known issues

Version 2.0.0 of the Splunk Add-on for Microsoft SQL Server has the following known issues.

If no issues appear below, no issues have yet been reported:

Date filed	Issue number	Description
2014-12-18	ADDON-2753, ADDON-8229	Error in opening perfmon with regex object (SQLServer|MSSQL*) from data inputs UI

Third-party software attributions

Version 2.0.0 of the Splunk Add-on for Microsoft SQL Server does not incorporate any third-party components or libraries.

Version 1.4.0

Version 1.4.0 of the Splunk Add-on for Microsoft SQL Server was released on August 2, 2018.

About this release

Version 1.4.0 of the Splunk Add-on for Microsoft SQL Server is compatible with the following software, CIM versions, and platforms.

Splunk platform versions	6.6.x, 7.0.x, 7.1.x, 7.2.x, 8.0
Splunk DB Connect	2.4.1, 3.1.3

CIM	4.11
Platforms	Windows for local data collection on MS SQL Server, platform independent otherwise
Vendor Products	Microsoft SQL Server 2008 R2 Enterprise, Microsoft SQL Server 2012 Enterprise, Microsoft SQL Server 2014 Enterprise, Microsoft SQL Server 2016 Enterprise

New Features

- Support for Microsoft SQL Server 2016
- Compatibility with Splunk Enterprise 7+

Known issues

Version 1.4.0 of the Splunk Add-on for Microsoft SQL Server has the following known issues.

If no issues appear below, no issues have yet been reported:

Date filed	Issue number	Description
2020-10-22	ADDON-30436	'Additional_Information' field is not getting extracted properly for the 'mssql:audit' sourcetype
2020-09-18	ADDON-29421	Removed incorrect field mapping of serial_num in sourcetype = mssql:transaction:dm_tran_locks
2014-12-18	ADDON-2753, ADDON-8229	Error in opening perfmon with regex object (SQLServer|MSSQL*) from data inputs UI

Third-party software attributions

Version 1.4.0 of the Splunk Add-on for Microsoft SQL Server does not incorporate any third-party components or libraries.

Version 1.3.0

Version 1.3.0 of the Splunk Add-on for Microsoft SQL Server is compatible with the following software, CIM versions, and platforms:

Splunk platform versions	6.4 and later
Splunk DB Connect	2.4.1, 3.1.3 (versions 1.x and 3.0 are not supported)
CIM	4.1 and later
Platforms	Windows for local data collection on MS SQL Server, platform independent otherwise
Vendor Products	Microsoft SQL Server versions 2008 R2 Enterprise, 2012 Enterprise, and 2014 Enterprise

Upgrade guide

Version 1.2.0 of the Splunk Add-on for Microsoft SQL Server includes a new `default\sqlserver_dbx2.conf` file. If you are using DB Connect v2 and want to use this add-on with Splunk IT Service Intelligence, follow the directions in Configure DB Connect v2 inputs for the Splunk Add-on for Microsoft SQL Server to override your existing inputs with the new ones provided in this version of the `default\sqlserver_dbx2.conf` template file. If you are using the configuration files, use the `default\sqlserver_dbx2.conf` as a template to update your `splunk_app_db_connect\local\inputs.conf`. If you are using the DB Connect GUI, refer to `default\sqlserver_dbx2.conf` for the source types and query statements.

DB Connect v1 is not supported for collecting data with this add-on for use in Splunk IT Service Intelligence. If you want to use this add-on with Splunk IT Service Intelligence, upgrade to DB Connect v2.

Features

Version 1.3.0 of the Splunk Add-on for Microsoft SQL Server provides added support for DB Connect 3.1, which significantly streamlines the configuration process of database inputs.

Fixed issues

Version 1.3.0 of the Splunk Add-on for Microsoft SQL Server has no fixed issues.

Known issues

Version 1.3.0 of the Splunk Add-on for Microsoft SQL Server has no known issues.

Third-party software attributions

Version 1.3.0 of the Splunk Add-on for Microsoft SQL Server does not incorporate any third-party components or libraries.

Version 1.2.0

Version 1.2.0 of the Splunk Add-on for Microsoft SQL Server was released on April 1, 2016. Version 1.2.0 of the Splunk Add-on for Microsoft SQL Server is compatible with the following software, CIM versions, and platforms:

Splunk platform versions	6.1 and later
CIM	4.1 and later
Platforms	Windows for local data collection on MS SQL Server, platform independent otherwise
Vendor Products	Microsoft SQL Server versions 2008 R2 Enterprise, 2012 Enterprise, and 2014 Enterprise

Upgrade guide

Version 1.2.0 of the Splunk Add-on for Microsoft SQL Server includes a new `default\sqlserver_dbx2.conf` file. If you are using DB Connect v2 and want to use this add-on with Splunk IT Service Intelligence, follow the directions in Configure DB Connect v2 inputs for the Splunk Add-on for Microsoft SQL Server to override your existing inputs with the new ones provided in this version of the `default\sqlserver_dbx2.conf` template file. If you are using the configuration files, use the `default\sqlserver_dbx2.conf` as a template to update your `splunk_app_db_connect\local\inputs.conf`. If you are using the DB Connect GUI, refer to `default\sqlserver_dbx2.conf` for the source types and query statements.

DB Connect v1 is not supported for collecting data with this add-on for use in Splunk IT Service Intelligence. If you want to use this add-on with Splunk IT Service Intelligence, upgrade to DB Connect v2.

Features

Version 1.2.0 of the Splunk Add-on for Microsoft SQL Server has the following new features:

Date	Issue number	Description
2016-03-09	ADDON-7327	Support for the IT Service Intelligence Database module, including new source types, new lookup file, and new SQL queries for DB Connect v2.

Date	Issue number	Description

Fixed issues

Version 1.2.0 of the Splunk Add-on for Microsoft SQL Server has no fixed issues.

Known issues

Version 1.2.0 of the Splunk Add-on for Microsoft SQL Server has the following known issues:

Date filed	Issue number	Description
2016-03-30	ADDON-2764	Data type RAW (8 byte) not supported due to limitation of DB Connect v.2.0.0. As a result some fields have a value of '## NOT SUPPORTED TYPE ##'.
2015-01-08	ADDON-2764	Incorrect line breaking and/or some fields for audit and trace log events are missing in indexed events when using DB Connect v1. Workaround: edit `dbx\local\inputs.conf` to include `SHOULD_LINEMERGE=true` in the stanzas for the affected inputs.
2015-02-04	ADDON-3131	Change to non-deprecated method for pulling trace log files from SQL Server because fn_trace_gettable is EOL.
2014-12-18	ADDON-2753	Error in creating performance monitor inputs with regex objects (SQLServer|MSSQL[^:]*) from data inputs UI. Workaround: configure all performance monitoring inputs via the `inputs.conf` file.

Note that there is a known issue in DB Connect 3 to support a new installation of Splunk Add-on for Microsoft SQL Server. See release notes for DB Connect 3 for details.

Third-party software attributions

Version 1.2.0 of the Splunk Add-on for Microsoft SQL Server does not incorporate any third-party components or libraries.

Version 1.1.0

Version 1.1.0 of the Splunk Add-on for Microsoft SQL Server has the same compatibility specifications as Version 1.2.0.

Features

Version 1.1.0 of the Splunk Add-on for Microsoft SQL Server has the following new features.

Date	Issue number	Description
2015-10-09	ADDON-2921	Support for Splunk DB Connect v2.
2015-10-09	ADDON-3734	Support for Microsoft SQL Server 2014.

Fixed issues

Version 1.1.0 of the Splunk Add-on for Microsoft SQL Server fixes the following issues.

Date fixed	Issue number	Description
2015-10-08	ADDON-5975	Stanzas have unnecessary wildcarding, contrary to best practices.
2015-10-08	ADDON-5978	Sourcetypes in eventtypes.conf should be capitalization-unified with real sourcetypes' names.

Date fixed	Issue number	Description
2015-09-23	ADDON-3270	Console startup errors when DB Connect is not present.

Known issues

Version 1.1.0 of the Splunk Add-on for Microsoft SQL Server has the following known issues.

Date filed	Issue number	Description
2015-01-08	ADDON-2764	Incorrect line breaking and/or some fields for audit and trace log events are missing in indexed events when using DB Connect v1. Workaround: edit `dbx\local\inputs.conf` to include `SHOULD_LINEMERGE=true` in the stanzas for the affected inputs.
2015-02-04	ADDON-3131	Change to non-deprecated method for pulling trace log files from SQL Server because fn_trace_gettable is EOL.
2014-12-18	ADDON-2753	Error in creating performance monitor inputs with regex objects (SQLServer|MSSQL[^:]*) from data inputs UI. Workaround: configure all performance monitoring inputs via the `inputs.conf` file.

Third-party software attributions

Version 1.1.0 of the Splunk Add-on for Microsoft SQL Server does not incorporate any third-party components or libraries.

Version 1.0.0

Version 1.0.0 of the Splunk Add-on for Microsoft SQL Server is compatible with the following software, CIM versions, and platforms.

Splunk platform versions	6.1 and later
CIM	4.1 and later
Platforms	Windows
Vendor Products	Microsoft SQL Server versions 2008 R2 Enterprise and 2012 Enterprise

New features

Version 1.0.0 of the Splunk Add-on for Microsoft SQL Server has the following new features.

Date	Issue number	Description
12/12/14	ADDON-211	CIM-compliant data collection of performance metrics and security events from Microsoft SQL Server using log files, audit trace files, and Splunk DB Connect.

Known issues

Version 1.0.0 of the Splunk Add-on for Microsoft SQL Server has the following known issues.

Date	Issue number	Description
01/08/15	ADDON-2764	Incorrect line breaking and/or some fields for audit and trace log events are missing in indexed events. Workaround: edit `dbx\local\inputs.conf` to include `SHOULD_LINEMERGE=true` in the stanzas for the affected inputs.
02/04/15	ADDON-3131	Change to non-deprecated method for pulling trace log files from SQL Server because fn_trace_gettable is EOL.

Date	Issue number	Description
12/18/14	ADDON-2753	Error in creating performance monitor inputs with regex objects (SQLServer|MSSQL[^:]*) from data inputs UI. Workaround: configure all performance monitoring inputs via the `inputs.conf` file.

Third-party software attributions

Version 1.0.0 of the Splunk Add-on for Microsoft SQL Server does not incorporate any third-party components or libraries.

Hardware and software requirements for the Splunk Add-on for Microsoft SQL Server

Splunk admin requirements

To install and configure the Splunk Add-on for Microsoft SQL Server, you must be member of the `admin` or `sc_admin` role.

Splunk DB Connect

This add-on requires Splunk DB Connect to collect data from data from trace logs, audit logs, and Dynamic Management Views. This add-on supports versions 3.6.0 and later of DB Connect.

You can run Splunk DB Connect on a search head or a heavy forwarder and collect data remotely from your Microsoft SQL Server. See Prerequisites in *Deploy and Use Splunk DB Connect* for setup requirements.

Microsoft SQL Server setup requirements

To collect Microsoft SQL Server error log, agent log, and performance data, install a Windows Splunk universal forwarder directly on each machine running Microsoft SQL Server.

Splunk platform requirements

This add-on must be installed on Windows instance of the Splunk platform for local monitoring. The add-on is platform independent for index and search time functions, as well as for DB Connect inputs.

Because this add-on runs on the Splunk platform, all of the system requirements apply for the Splunk software that you use to run this add-on.

- For Splunk Enterprise system requirements: see System Requirements in the Splunk Enterprise *Installation Manual*.
- For Splunk Light system requirements: see System Requirements in the Splunk Light *Installation Manual*.
- If you are managing on-premises forwarders to get data into Splunk Cloud, see System Requirements in the Splunk Enterprise *Installation Manual*, which includes information about forwarders.

For information about installation locations and environments, see Install the Splunk Add-on for Microsoft SQL Server.

Installation overview for the Splunk Add-on for Microsoft SQL Server

To install and configure this add-on, follow these steps:

1. Create the audit objects in Microsoft SQL Server to enable monitoring of SQL server audit data. If you do not want to index server audit data, you can skip this step without affecting your other inputs.
2. Install this add-on.
3. Configure your monitor inputs and Windows Performance Monitoring inputs.
4. configure DB Connect v3 inputs.

Installation and Configuration

Create audit objects in Microsoft SQL Server for the Splunk Add-on for Microsoft SQL Server

The Splunk Add-on for Microsoft SQL Server includes support for monitoring audit data from Microsoft SQL Server. For more information, search for "SQL Server Audit (Database Engine)" on the MSDN web site. By default, auditing is disabled in SQL Server, so you must create the audit objects in your SQL Server instance in order for the Splunk platform to ingest this data. If you skip this step, the add-on does not collect audit log data, but the other inputs still function.

You can configure auditing in Microsoft SQL Server at the server level or at the database level. Manage the audit level by configuring audit action items that target server-level operations, database-level operations, or individual operations on a database table, view, or stored procedure. See "SQL Server Audit Action Groups and Actions" in the Microsoft SQL Server documentation for a full guide covering how to set up audit action groups and actions.

> The more types of audit specifications you monitor, the larger the audit events indexed by the Splunk platform.

Create audit objects and specifications using SQL Server Management Studio or Transact-SQL. See "Create a Server Audit and Server Audit Specification" in the Microsoft SQL Server documentation for a step-by-step guide.

The following examples demonstrate how to create and enable audit objects at different possible levels.

Example of creating an audit object at the server level

Create the C:\SQLAudit directory first.

1. Create a server-level audit object `MSSQL_Server_Audit` with a file path `C:\SQLAudit`.

   ```
   USE master ;

   -- Create the server audit.

   CREATE SERVER AUDIT MSSQL_Server_Audit TO FILE ( FILEPATH = 'C:\\SQLAudit' ) ;

   -- Enable the server audit.

   ALTER SERVER AUDIT MSSQL_Server_Audit WITH (STATE = ON) ;
   ```
2. Create an Audit Specification `MSSQL_Server_Specification` in the master database and attach it to the audit object. This Audit Specification audits the following groups. `APPLICATION_ROLE_CHANGE_PASSWORD_GROUP`, `BROKER_LOGIN_GROUP`, `DATABASE_CHANGE_GROUP`, `DATABASE_LOGOUT_GROUP`.

   ```
   USE master;

   CREATE SERVER AUDIT SPECIFICATION MSSQL_Server_Specification
   FOR SERVER AUDIT MSSQL_Server_Audit
   ADD (APPLICATION_ROLE_CHANGE_PASSWORD_GROUP), ADD (BROKER_LOGIN_GROUP),
   ADD (DATABASE_CHANGE_GROUP), ADD (DATABASE_LOGOUT_GROUP) WITH (STATE = ON) ;
   ```

Example of creating an audit object at the database level

1. Create a database-level audit object `MSSQL_Database_Audit` with a file path `C:\SQLAudit`.

```
USE master ;

CREATE SERVER AUDIT MSSQL_Database_Audit TO FILE ( FILEPATH = 'C:\\SQLAudit' ) ;

-- Enable the server audit.

ALTER SERVER AUDIT MSSQL_Database_Audit WITH (STATE = ON) ;
```

2. Create the Audit Specification `MSSQL_Database_Specification` in the master database and attach it to the audit object. This Audit Specification audits the following groups. `APPLICATION_ROLE_CHANGE_PASSWORD_GROUP`, `AUDIT_CHANGE_GROUP` and `DATABASE_LOGOUT_GROUP`.

```
USE master;

CREATE DATABASE AUDIT SPECIFICATION MSSQL_Database_Specification
FOR SERVER AUDIT MSSQL_Database_Audit
ADD (APPLICATION_ROLE_CHANGE_PASSWORD_GROUP),
ADD (AUDIT_CHANGE_GROUP), ADD (DATABASE_LOGOUT_GROUP) WITH (STATE = ON) ;
```

Example of creating an audit object for a table

1. Create a database-level audit object `MSSQL_Table_Audit` with a file path `C:\SQLAudit`.

```
USE master ;

-- Create the server audit.

CREATE SERVER AUDIT MSSQL_Table_Audit TO FILE ( FILEPATH = 'C:\\SQLAudit' ) ;

-- Enable the server audit.

ALTER SERVER AUDIT MSSQL_Table_Audit WITH (STATE = ON) ;
```

2. Create Table Audit Specification `MSSQL_Table_Specification` which audits the `update` action on table `Payment` in the `HumanResource` database.

```
USE HumanResource;

-- Create the database audit specification.

CREATE DATABASE AUDIT SPECIFICATION MSSQL_Table_Specification
FOR SERVER AUDIT MSSQL_Table_Audit
ADD (UPDATE ON Payment BY dbo ) WITH (STATE = ON) ;
```

Install the Splunk Add-on for Microsoft SQL Server

To install the Splunk Add-on for Microsoft SQL Server, perform the following steps:

1. Get the Splunk Add-on for Microsoft SQL Server by downloading it from http://splunkbase.splunk.com/app/2648 or browsing to it using the app browser within Splunk Web.
2. Determine where and how to install this add-on in your deployment, using the tables on this page.
3. Perform any prerequisite steps before installing, if required and specified in the tables below.
4. Complete your installation.

For information on upgrading to the newest version of this add-on, see the Upgrade the Splunk Add-on for Microsoft SQL Server topic in this manual.

If you need step-by-step instructions on how to install an add-on in your specific deployment environment, see the installation walkthroughs section at the bottom of this page for links to installation instructions specific to a single-instance deployment, distributed deployment, Splunk Cloud, or Splunk Light.

Distributed deployment

Use the tables below to determine where and how to install this add-on in a distributed deployment of Splunk Enterprise or any deployment for which you are using forwarders to get your data in. Depending on your environment, your preferences, and the requirements of the add-on, you may need to install the add-on in multiple places.

Where to install this add-on

This table provides a quick reference for installing this add-on to a distributed deployment of Splunk Enterprise.

Splunk instance type	Supported	Required	Comments
Search Heads	Yes	Yes	Install this add-on to all search heads where Microsoft SQL Server knowledge management is required.
Indexers	Yes	No	Not required, because this add-on does not include any index-time operations.
Heavy Forwarders	Yes	No	To collect dynamic management view data, trace logs, and audit logs, you must use Splunk DB Connect on a search head or heavy forwarder. The remaining data types support using a universal or light forwarder installed directly on the machines running MS SQL Server.
Universal Forwarders	Yes	No	To collect dynamic management view data, trace logs, and audit logs, you must use Splunk DB Connect on a search head or heavy forwarder. The remaining data types support file monitoring using a universal or light forwarder installed directly on the machines running MS SQL Server.

Distributed deployment feature compatibility

This table provides a quick reference for the compatibility of this add-on with Splunk distributed deployment features.

Distributed deployment feature	Supported	Comments
Search Head Clusters	Yes	You can install this add-on on a search head cluster for all search-time functionality. Before installing this add-on to a cluster, remove the `default\inputs.conf` file from the add-on package. If you run your DB Connect inputs from your search head cluster captain, you can keep the `default\inputs.conf` file on your search heads for reference. See the Configure DB Connect version 3.6.x inputs for the Splunk Add-on for Microsoft SQL Server topic in this manual to learn how to enable DB Connect inputs using Splunk Web.
Indexer Clusters	Yes	Before installing this add-on to a cluster, remove the `default\inputs.conf` file from the add-on package.
Deployment Server	Conditional	Supported for deploying the configured add-on to multiple universal forwarders for local data collection via file monitoring and Windows performance monitoring. Not supported for DB Connect inputs.

Installation walkthrough

See Installing add-ons in *Splunk Add-Ons* for detailed instructions describing how to install a Splunk add-on in the following deployment scenarios:

- single-instance Splunk Enterprise
- distributed Splunk Enterprise
- Splunk Cloud
- Splunk Light

Configure monitor inputs and Windows Performance Monitoring inputs for the Splunk Add-on for Microsoft SQL Server

The Splunk Add-on for Microsoft SQL Server allows you to collect a variety of log and performance data from your Microsoft SQL Server instances. The `default\inputs.conf` file has a complete set of input stanzas that you can use as a basis for your local configurations.

The following configuration instructions assume that you have installed the Splunk Add-on for Microsoft SQL Server on forwarders installed directly on your machines running Microsoft SQL Server.

Prepare your `local\inputs.conf` file

1. Open `%SPLUNK_HOME%\etc\apps\Splunk_TA_microsoft-sqlserver\default\inputs.conf`.
2. Copy the contents to `%SPLUNK_HOME%\etc\apps\Splunk_TA_microsoft-sqlserver\local\inputs.conf`

Configure error and agent log file monitoring

Configure monitoring of error and agent logs using Splunk File Monitoring.

Prerequisite: The server agent log file may not exist if the SQL Server Agent Service has never started. Be sure to start the agent first before attempting to configure a monitor input.

1. If necessary, edit the file path of the inputs to match the actual location of the log files in your environment.
2. Enable the inputs by changing `disabled = 1` to `disabled = 0`.

It is also possible to configure these inputs via Splunk Web on your heavy forwarder. For more information about configuring file monitoring on Splunk Web, see Monitor files and directories with Splunk Web in the *Getting Data In* manual, part of the Splunk Enterprise documentation.

Configure performance monitoring

Configure monitoring of your Microsoft SQL Server instances and the Windows systems running them using Windows Performance Monitoring.

1. Open `%SPLUNK_HOME%\etc\apps\Splunk_TA_microsoft-sqlserver\local\inputs.conf`.
2. Enable performance monitoring for the tasks you are interested in by changing `disabled = 1` to `disabled = 0` for those stanzas.

For more details about each performance monitoring task, refer to the Windows and SQL Server performance data section of the source types reference page.

Configure DB Connect version 3.6.x inputs for the Splunk Add-on for Microsoft SQL Server

To gather trace logs, audit logs, and data from Dynamic Management Views, the Splunk Add-on for Microsoft SQL Server leverages Splunk DB Connect. Follow the instructions that correspond to the version of DB Connect that you have installed. This topic presents the instructions for DB Connect Version 3.6 and above.

To prepare your environment and configure your inputs, follow these steps.

1. Set up the database connection
2. Configure the inputs
3. Adjust your auto KV extraction settings
4. Configure the database server lookup

Set up the database connection

Setting up the database connection involves three steps:

1. Download and install the Microsoft JDBC driver for SQL Server.
2. Create an identity in Splunk platform.
3. Use the Splunk DB Connect GUI to create a database connection or use `db_connections.conf` to create a database connection.

Download and install the Microsoft JDBC driver for SQL Server

To enable Microsoft SQL Server connections, download and install the Microsoft JDBC Driver for SQL Server as described in the Install database drivers section of the *Deploy and Use Splunk DB Connect* manual.

Create an identity in the Splunk platform

1. Restart the Splunk platform instance.
2. Create an identity for establishing a connection to the database. Make sure the user for this identity has the system role. You can use a username and password for authentication, or Windows Authentication. However, using DB Connect version 3.1 with Windows Authentication and the JDBC driver for SQL Server requires additional steps. See Can't use Windows authentication for Microsoft SQL Server with Microsoft JDBC Driver for SQL Server in the DB Connect manual for more information.
3. Next, you need to create a database connection to the SQL Server using either the Splunk DB Connect GUI or the `db_connections.conf` file as described in the following sections.

Use the Splunk DB Connect GUI to create a database connection

To create a database connection to the SQL Server database using the Splunk DB Connect GUI, refer to Create and manage database connections in the Splunk DB Connect manual for step-by-step instructions.

Enter the following parameters:

Parameter	Value
Connection Name	Enter a unique connection name.
Identity	Use the identity you created above.
Connection Type	Choose **MS-SQL Server Using MS Generic Driver** or **MS-SQL Server Using jTDS Driver** based on which driver you are using.
Host	Enter the host IP address where the SQL Server database is running.
Port	The default port for SQL Server database is 1433.
Default Database:	Enter the database name on SQL Server.

Use `db_connections.conf` to create a database connection

If you do not want to use the DB Connect GUI, you can create a database connection to the SQL Server database using the `db_connections.conf` file.

1. Create a file called `db_connections.conf` in the `%SPLUNK_HOME%\etc\apps\splunk_app_db_connect\local` directory.
2. Copy the stanza below to `db_connections.conf` and edit the values of each field to reflect your production environment.

```
<connection name>
connection_type = <connection type: generic_mssql or mssql>
database = <database name>
host = <host or ip address of the SQL Server database>
identity = <identity name used for the connection>
jdbcUseSSL = <enable SSL>
port = <network port of the SQL Server database>
```

When you create a database connection object for your Microsoft SQL Server, select the appropriate database and driver from the Database Types pop-up menu. There are two options: **MS-SQL Server Using MS Generic Driver** if you downloaded the Microsoft driver, or M**S-SQL Server Using jTDS Driver** if you are using the open source jTDS driver.

Configure database inputs using the Splunk DB Connect GUI

Refer to Create and manage database inputs in the Splunk DB Connect manual for step by step instructions configuring your database inputs in the GUI.

If you want to create Microsoft SQL Server input, choose the template created for **Splunk Add-on for Microsoft SQL Server** under **Template** field of DB Connect. .

Adjust your auto KV extraction settings

Some source types, such as `mssql:execution:dm_exec_query_stats` retrieve fields with multiple lines. To ensure that your fields show the full values that you expect, adjust your KV extraction settings.

1. Open `%SPLUNK_HOME%\etc\system\local\limits.conf`.
2. Add or change this stanza:

```
[kv]
maxchars = 20480
```

Configure the database server lookup

Supply the `host` and `port` values for each of your database servers. This step is required to integrate with Splunk IT Service Intelligence.

1. On each of your search heads, open `%SPLUNK_HOME%\etc\apps\Splunk_TA_microsoft-sqlserver\lookups\sqlserver_host_dbserver_lookup.csv`.
2. Edit this file to include correct `host` and `port` values for each of the `database_server` in your event data.
3. Save the file.
4. Restart the search head.

Configuring MSSQL to collect data through Windows AD

All the data obtained via DB Connect are from Dynamic Management View or Dynamic Management Function including both server-scoped and database-scoped. This requires a user with both `VIEW SERVER STATE` permission and `VIEW DATABASE STATE` permission.

For example, If a user called 'splunk' is created, admin user needs to grant the permission to the user using the following SQL command:

```
GRANT VIEW SERVER STATE TO splunk; GRANT VIEW DATABASE STATE TO splunk;
```

For more information, check the oficial SQL Server Doc

Configure DB Connect v2 inputs for the Splunk Add-on for Microsoft SQL Server

To gather trace logs, audit logs, and data from Dynamic Management Views, the Splunk Add-on for Microsoft SQL Server leverages Splunk DB Connect. Follow the instructions that correspond to the version of DB Connect that you have installed. This topic presents the instructions for DB Connect Version 2.x.

To prepare your environment and configure your inputs, follow these steps.

1. Set up the database connection
2. Configure the inputs
3. Adjust your auto KV extraction settings
4. Configure the database server lookup

Set up the database connection

Setting up the database connection involves three steps:

1. Download and install the Microsoft JDBC driver for SQL Server.
2. Create an identity in Splunk platform.
3. Use the Splunk DB Connect GUI to create a database connection or use `db_connections.conf` to create a database connection.

Download and install the Microsoft JDBC driver for SQL Server

To enable Microsoft SQL Server connections, download and install the Microsoft JDBC Driver for SQL Server as described in the Install database drivers section of the *Deploy and Use Splunk DB Connect* manual.

Create an identity in the Splunk platform

1. Restart the Splunk platform instance.
2. Create an identity for establishing a connection to the database. Make sure the user for this identity has the system role. You can use a username and password for authentication, or Windows Authentication. However, using DB Connect version 2.x with Windows Authentication and the JDBC driver for SQL Server requires additional steps. See Cannot connect to Microsoft SQL server in the DB Connect manual for more information.
3. Next, you need to create a database connection to the SQL Server using either the Splunk DB Connect GUI or the `db_connections.conf` file as described in the following sections.

Use the Splunk DB Connect GUI to create a database connection

To create a database connection to the SQL Server database using the Splunk DB Connect GUI, refer to Create and manage database connections in the Splunk DB Connect manual for step-by-step instructions.

Enter the following parameters:

Parameter	Value
Connection Name	Use `SQLServer` for the connection name. If you prefer to choose a different connection name, you need to manually edit your local `inputs.conf` file later to specify a non-default name.
Identity	Use the identity you created above.
App	Use the default app, Splunk DB Connect V2.
Port	The default port for SQL Server database is 1433.
Host	Enter the host IP address where the SQL Server database is running.
Database Types	Choose **MS-SQL Server Using MS Generic Driver** or **MS-SQL Server Using jTDS Driver** based on which driver you are using.
Default Database:	Enter the database name on SQL Server.

Use `db_connections.conf` to create a database connection

If you do not want to use the DB Connect GUI, you can create a database connection to the SQL Server database using the `db_connections.conf` file.

1. Create a file called `db_connections.conf` in the `%SPLUNK_HOME%\etc\apps\splunk_app_db_connect\local` directory.
2. Copy the stanza below to `db_connections.conf` and edit the values of each field to reflect your production environment.

```
<connection name>
connection_type = <connection type: generic_mssql or mssql>
database = <database name>
host = <host or ip address of the SQL Server database>
identity = <identity name used for the connection>
jdbcUseSSL = <enable SSL>
port = <network port of the SQL Server database>
```

When you create a database connection object for your Microsoft SQL Server, select the appropriate database and driver from the Database Types pop-up menu. There are two options: **MS-SQL Server Using MS Generic Driver** if you downloaded the Microsoft driver, or M**S-SQL Server Using jTDS Driver** if you are using the open source jTDS driver.

Configure the inputs

To configure the inputs, you can copy the input template provided in the add-on to your local `inputs.conf` file and enable the inputs that you want to collect there. Alternatively, you can configure the DB Connect inputs manually using the DB Connect GUI.

Use `inputs.conf` to configure your database inputs

1. Copy the contents of `%SPLUNK_HOME%\etc\apps\Splunk_TA_microsoft-sqlserver\default\sqlserver_dbx2.conf` to `%SPLUNK_HOME%\etc\apps\splunk_app_db_connect\local\inputs.conf`.
2. Change `disabled = 1` to `disabled = 0` in the input stanzas you want to use.
3. If you selected a custom connection name other than `SQLServer`, change that parameter here in each stanza to match the connection name that you configured in `db_connections.conf` or via the GUI.
4. Change the file path for `[mi_input://mssql:trclog]` and `[mi_input://mssql:audit]` based on the paths that you specified when you created the audit objects in Microsoft SQL Server.
5. (Optional) Select a custom index. The default is main.
6. Restart the Splunk platform instance for your changes to take effect.

Use the Splunk DB Connect GUI to configure your database inputs

Refer to Create and manage database inputs in the Splunk DB Connect manual for step by step instructions for using the GUI to configure your database inputs.

The following example shows the configuration instructions for the `sys.processes` Dynamic Management View. Refer to `default/sqlserver_dbx2.conf` for the source types and query statements.

Parameter	Setting Value
Status	Enabled
Name	mssql:processes
Description	
App	Splunk DB Connect v2
Connection	Enter the database connection name you created when you set up the database connection. These instructions recommended `SQLServer`.
Query Mode	Advanced Query Mode
Query Statement	`SELECT a.*, b.name,CONVERT(varchar(128),SERVERPROPERTY('ServerName')) AS ServerName, db_name() AS DatabaseName FROM sys.sysprocesses a JOIN sys.databases b ON a.dbid = b.database_id`
Type	Batch Input
Max Rows to Retrieve	10000
Timestamp	Current Index Time

Parameter	Setting Value
Output Timestamp Format	YYYY-MM-dd HH:mm:ss
Execution Frequency	300
Source	dbx2
Sourcetype	`mssql:processes`
Index	main
Select Resource Pool	local

Adjust your auto KV extraction settings

Some source types, such as `mssql:execution:dm_exec_query_stats` retrieve fields with multiple lines. To ensure that your fields show the full values that you expect, adjust your KV extraction settings.

1. Open `%SPLUNK_HOME%\etc\system\local\limits.conf`.
2. Add or change this stanza:

```
[kv]
maxchars = 20480
```

Saved searches for the Splunk Add-on for Microsoft SQL Server

The TA provides a saved search named *SQL Server - Latest Process Information*. This saved search gets the latest fetching data for sourcetype mssql:processes.

Upgrade the Splunk Add-on for Microsoft SQL Server

Use the Install the Splunk Add-on for Microsoft SQL Server topic in this manual to upgrade to the latest version of the Splunk Add-on for Microsoft SQL Server.

Update your inputs with updated templates

Starting in version 2.0.0 of the Splunk Add-on for Microsoft SQL Server, SQL queries in the `db_template` file extract the `host` and `port` fields from the events and no longer provide these fields in the lookup, by default. No change is required for file monitoring and windows performance monitoring inputs.

SQL queries are updated for the following templates:

- `mssql:instance`
- `mssql:os:dm_os_performance_counters`
- `mssql:table`
- `mssql:user`
- `mssql:os:dm_os_sys_info`
- `mssql:instancestats`
- `mssql:execution:dm_exec_sessions`
- `mssql:transaction:dm_tran_locks`

- `mssql:execution:dm_exec_query_stats`

If you use any of these templates, update your inputs with the updated template by performing the following steps:

1. In a Splunk instance with Splunk DB Connect installed, open Splunk DB Connect.
2. Navigate to the the **Inputs** tab.
3. Navigate an input that uses an updated template. Use the **Template** column to filter your inputs by template.
4. In the **Actions** column, click **Edit**. The **Edit input** window appears.
5. In the **Edit input** window, click the **Refresh** icon on the right side
6. Once the query is reloaded, click the **Execute SQL** button to execute the updated query.
7. Click **Next**.
8. Save your changes.
9. Repeat process for each input that uses an updated template.

Troubleshooting

Troubleshoot the Splunk Add-on for Microsoft SQL Server

General troubleshooting

For helpful troubleshooting tips that you can apply to all add-ons, see Troubleshoot add-ons in *Splunk Add-ons*. For additional resources, see Support and resource links for add-ons in *Splunk Add-ons*.

Verify data collection

To check that the Splunk platform is collecting the data that you expect, use this search command to list the indexed source types:

```
| metadata index=main type=sourcetypes | fields sourcetype
```

You can find all possible source types contributed by this add-on listed on the source types page. If your search results are missing a source type that you intended to collect, verify that you have enabled that source type in `%SPLUNK_HOME%\etc\apps\Splunk_TA_microsoft-sqlserver\local\inputs.conf`, `%SPLUNK_HOME%\etc\apps\splunk_app_db_connect\local\inputs.conf`.

Determine event sources

The add-on adds two fields at search time for each event that can be useful in determining the event source.

Field	How to determine the event source
`sqlserver_instance_name`	For events collected through file or performance monitoring, use the SQL Server instance name. For example: `default_instance` For events collected through Splunk DB Connect, use "DBConn:"+ connection_name. For example: `DBConn:sqlserver_default_connection`
`sqlserver_full_instance_name`	For events collected through file or performance monitoring, use host_name + "/"+sqlserver_instance_name. For example: `DBConn:sqlserver_default_connection` For events collected through Splunk DB Connect, usehost_name +"/"+"DBConn:"+ connection_name. For example: `WIN-J0NE2C7KVR9/DBconn:sqlserver_default_connection`

Eliminate white spaces in events

Some events from Microsoft SQL Server data contain extra white spaces caused by fixed sizes in SQL Server and the handling logic in Splunk DB Connect. Trim the white spaces using the search below:

```
sourcetype = [insert source type] | rex mode=sed "s/\s{2,}//g"
```

Line breaking and missing field issues for `mssql:audit` and `mssql:trclog` data collected with DB Connect v1

Due to limitations in Splunk DB Connect v1, you might experience line breaking issues or missing fields in your audit and trace logs. You can work around the issue by adding `SHOULD_LINEMERGE = true` in your `mssql:audit` and `mssql:trclog` stanzas in `%SPLUNK_HOME%\etc\apps\dbx\local\props.conf`.

Examine the buffer pool

If you want to see what objects and indexes are in the buffer pool, run the following SQL statements in DB query in Splunk DB Connect.

Note: Query statements in the DB connect conf files are limited to 128 characters, so this query must be run in Splunk Web.

```
select

       count(*)as cached_pages_count,

       obj.name as objectname,

       ind.name as indexname,

       obj.index_id as indexid

from sys.dm_os_buffer_descriptors as bd

    inner join

    (

        select       object_id as objectid,

                           object_name(object_id) as name,

                           index_id,allocation_unit_id

        from sys.allocation_units as au

            inner join sys.partitions as p

                on au.container_id = p.hobt_id

                    and (au.type = 1 or au.type = 3)

        union all

        select       object_id as objectid,

                           object_name(object_id) as name,

                           index_id,allocation_unit_id

        from sys.allocation_units as au

            inner join sys.partitions as p
```

```
                on au.container_id = p.partition_id

                    and au.type = 2

    ) as obj

        on bd.allocation_unit_id = obj.allocation_unit_id

left outer join sys.indexes ind

  on  obj.objectid = ind.object_id

 and  obj.index_id = ind.index_id

where bd.database_id = db_id()

  and bd.page_type in ('data_page', 'index_page')

group by obj.name, ind.name, obj.index_id

order by cached_pages_count desc
```

Understand the missing records_affected field

If you are using Microsoft SQL Server 2008 R2, the add-on does not provide the `records_affected` field for events in the source type `mssql:execution:dm_exec_query_stats`. The add-on cannot supply this field because the add-on derives this field from the `last_rows` column in the table `sys.dm_exec_query_stats`, but Microsoft SQL Server 2008 R2 does not include this column. The absence of this field does not affect the mapping of any other fields.

Reference

Lookups for the Splunk Add-on for Microsoft SQL Server

Starting with version 3.0.0 of the Splunk Add-on for Microsoft SQL Server, the lookup is not required, as the port field is included in the raw event. To get the port field from previously indexed events, perform the following steps to update the lookup.

Supply the port values for each of your database servers. This step is required to integrate with Splunk IT Service Intelligence.

1. On your search head, navigate to `%SPLUNK_HOME%\etc\apps\Splunk_TA_microsoft-sqlserver\lookups\`.
2. Open `sqlserver_host_dbserver_lookup.csv` with a text editor.
3. Edit the file to include the correct port values for each `database_instance` in your event data. For example, see the following table:

database_instance	port
MSSQL-19	1433

4. Save the file.
5. Restart the search head.
6. Repeat for each search head.