Integrate Microsoft SQL Server with EventTracker
EventTracker v9.0 and above

Publication Date: February 4, 2020
Abstract

This guide provides instructions to configure Microsoft SQL Server auditing and forward relevant events to EventTracker.

Scope

The configurations detailed in this guide are consistent with EventTracker version 9.x and later, Microsoft SQL Server 2012 and later Edition.

Audience

Administrators who want to monitor the Microsoft SQL Server using EventTracker.
Table of Contents

1. Introduction ........................................................................................................................................... 4

2. Server audit specifications ...................................................................................................................... 4
   2.1 Prerequisites ....................................................................................................................................... 5
   2.2 Enabling logging for logins ............................................................................................................... 5
   2.3 Enabling server auditing .................................................................................................................. 6
      2.3.1 Creating audits .......................................................................................................................... 7
      2.3.2 Creating server audit specifications ......................................................................................... 8

3. Extended events ..................................................................................................................................... 17
   3.1 Prerequisites ..................................................................................................................................... 17
   3.2 Creating extended event session ................................................................................................... 17
   3.3 Parse extended event session log file ............................................................................................. 20

4. EventTracker Knowledge Package ......................................................................................................... 21
   4.1 Saved Searches ............................................................................................................................... 21
   4.2 Alerts ............................................................................................................................................... 23
   4.3 Reports ........................................................................................................................................... 24
   4.4 Dashboards ...................................................................................................................................... 39

5. Importing knowledge pack into EventTracker ....................................................................................... 42
   5.1 Categories ....................................................................................................................................... 43
   5.2 Alerts .............................................................................................................................................. 44
   5.3 Token Templates ............................................................................................................................. 45
   5.4 Flex Reports .................................................................................................................................... 47
   5.5 Knowledge Objects ......................................................................................................................... 48
   5.6 Dashboards ..................................................................................................................................... 50

6. Verifying knowledge pack in EventTracker ......................................................................................... 51
   6.1 Categories ....................................................................................................................................... 51
   6.2 Alerts .............................................................................................................................................. 52
   6.3 Token Templates ............................................................................................................................. 52
6.4 Flex Reports .......................................................................................................................... 52
6.5 Knowledge Objects .................................................................................................................. 53
6.6 Dashboards ............................................................................................................................. 53
1. Introduction

Microsoft SQL Server is a relational database management system with several features and services. With this coverage, there is a large surface area for attack and vulnerabilities.

SQL Server auditing is utilized to address requirements for compliance, analyze database actions to troubleshooting problems and investigate suspicious activity.

EventTracker can employ both server audit specifications and extended events to receive relevant events for auditing. Configuration techniques for both methodologies are shown below. Please configure any method of your choice in accordance with your infrastructure and audit requirements. Both techniques are compared below:

<table>
<thead>
<tr>
<th>Audit Types</th>
<th>Pro’s</th>
<th>Con’s</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Audit Specifications</strong> (available in Microsoft SQL Server 2008 or later)</td>
<td>• Alerts are received in real-time.</td>
<td>• Additional fields like client hostname, client application name is missing.</td>
</tr>
<tr>
<td></td>
<td>• Events are received in Windows Event Viewer.</td>
<td></td>
</tr>
<tr>
<td><strong>Extended Events</strong> (available in Microsoft SQL Server 2012 or later)</td>
<td>• Additionally, provides a client hostname, client application name and event category fields.</td>
<td>• Alerts are received with a maximum delay of two hours.</td>
</tr>
<tr>
<td></td>
<td>• Lightweight and utilizes a few performance resources.</td>
<td></td>
</tr>
</tbody>
</table>

EventTracker MSSQL reports provide information about database activities. By using these reports, we can examine the user login success and login failures for further investigation. These reports can track the database changes in the tables, views, procedures, triggers, schema and can track any SQL query errors.

Dashboards display a graphical representation of the database object changes and actions carried out on that object.

Through dashboards, we can also easily track multiple brute force login failures. Alerts trigger when a user performs any changes on the database, database view, schema, user management etc.

2. Server audit specifications

Auditing, an instance of a SQL Server or a SQL Server database involves tracking and logging events that occur on the system. **Below mentioned configuration must be applied on all workstations, where SQL audit is required.**
2.1 Prerequisites

- **Microsoft SQL Server 2008 or later** must be installed.
- **Microsoft SQL Management Studio** for the respective version must be installed.
- **EventTracker Agent 9.x or later** must be installed on the SQL SERVER workstation.

2.2 Enabling logging for logins

1. Open **Microsoft SQL management studio** with appropriate credentials.
2. In **Object Explorer**, right-click on the database server and select **Properties**.

3. In the **Properties** panel, select **Security** in **Select a page** section.
4. In **Login auditing**, select **Both failed and successful logins**.
5. The above configuration generates event id “18453” (login success) and “18456” (login failure).

Note: Login success events are very noisy, enable with caution.

2.3 Enabling server auditing

1. Open Microsoft SQL management studio with appropriate credentials.
2. In Object Explorer, expand the Security tab to view Audits and Server Audit Specifications options.
2.3.1 Creating audits

1. Right-click Audits to select New Audit...

2. In Audit Properties, provide appropriate audit name and set audit destination as application log. The configured Audit properties pane is shown below:
3. Click **OK** to apply settings.

### 2.3.2 Creating server audit specifications

1. Right-click **Server Audit Specifications** and select **New Server Audit Specification**...
2. In Server **Audit Specification Properties**, provide an appropriate **specification name** and choose an earlier created **audit name** from the drop-down menu.

3. In **Actions** pane, select the following specifications from **Audit Action Type** drop-down.

To improve performance, enable action types consistent with your audit requirements.

<table>
<thead>
<tr>
<th>SN</th>
<th>Audit Action Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>DATABASE_ROLE_MEMBER_CHANGE_GROUP</td>
<td>This generates events whenever a login is added to or removed from a database role.</td>
</tr>
<tr>
<td>2.</td>
<td>SERVER_ROLE_MEMBER_CHANGE_GROUP</td>
<td>This generates events whenever a login is added or removed from a fixed server role.</td>
</tr>
<tr>
<td>3.</td>
<td>BACKUP_RESTORE_GROUP</td>
<td>This generates events whenever a backup or restore command is issued.</td>
</tr>
<tr>
<td>4.</td>
<td>AUDIT_CHANGE_GROUP</td>
<td>This generates events whenever any audit is created, modified or deleted.</td>
</tr>
<tr>
<td>5.</td>
<td>DATABASE_PERMISSION_CHANGE_GROUP</td>
<td>This generates events whenever a GRANT, REVOKE, or DENY is issued by any user in SQL Server for database-only events.</td>
</tr>
<tr>
<td>6.</td>
<td>SCHEMA_OBJECT_PERMISSION_CHANGE_GROUP</td>
<td>This generates events whenever a grant, deny, or revoke is issued for a schema object.</td>
</tr>
<tr>
<td>7.</td>
<td>SERVER_PERMISSION_CHANGE_GROUP</td>
<td>This generates events when a GRANT, REVOKE, or DENY is issued for permissions in the server scope.</td>
</tr>
<tr>
<td>8.</td>
<td>DATABASE_CHANGE_GROUP</td>
<td>This generates events when a database is created, altered, or dropped.</td>
</tr>
<tr>
<td></td>
<td>DATABASE_OBJECT_CHANGE_GROUP</td>
<td>This generates events when a CREATE, ALTER, or DROP statement is executed on database objects.</td>
</tr>
<tr>
<td>---</td>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>10.</td>
<td>DATABASE_PRINCIPAL_CHANGE_GROUP</td>
<td>This generates events when principals are created, altered, or dropped from a database.</td>
</tr>
<tr>
<td>11.</td>
<td>SCHEMA_OBJECT_CHANGE_GROUP</td>
<td>This generates events when a CREATE, ALTER, or DROP operation is performed on a schema.</td>
</tr>
<tr>
<td>12.</td>
<td>SERVER_OBJECT_CHANGE_GROUP</td>
<td>This generates events for CREATE, ALTER, or DROP operations on server objects.</td>
</tr>
<tr>
<td>13.</td>
<td>SERVER_PRINCIPAL_CHANGE_GROUP</td>
<td>This generates events when server principals are created, altered, or dropped.</td>
</tr>
<tr>
<td>14.</td>
<td>APPLICATION_ROLE_CHANGE_PASSWORD_GROUP</td>
<td>This generates events whenever a password is changed for an application role.</td>
</tr>
<tr>
<td>15.</td>
<td>LOGIN_CHANGE_PASSWORD_GROUP</td>
<td>This generates events whenever a login password is changed by ALTER LOGIN statement.</td>
</tr>
<tr>
<td>16.</td>
<td>DATABASE_OWNERSHIP_CHANGE_GROUP</td>
<td>This generates events when ALTER AUTHORIZATION statement is used to change the owner of a database.</td>
</tr>
<tr>
<td>17.</td>
<td>SCHEMA_OBJECT_OWNERSHIP_CHANGE_GROUP</td>
<td>This generates events when the permissions to change the owner of schema object (such as a table, procedure, or function) is checked.</td>
</tr>
<tr>
<td>18.</td>
<td>USER_CHANGE_PASSWORD_GROUP**</td>
<td>This generates events whenever the password of a contained database is changed.</td>
</tr>
</tbody>
</table>
Integrate Microsoft SQL Server with EventTracker

user is changed by using the ALTER USER statement.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>SUCCESSFUL_LOGIN_GROUP</td>
<td>This generates events whenever a successful logon is done</td>
</tr>
<tr>
<td>20</td>
<td>LOGOUT_GROUP</td>
<td>This generates events whenever a logout is done</td>
</tr>
<tr>
<td>21</td>
<td>FAILED_LOGON_GROUP</td>
<td>This generates events whenever a Logon failure happens</td>
</tr>
<tr>
<td>22</td>
<td>SERVICE_STATE_CHANGE_GROUP</td>
<td>This generates events whenever any SQL service is stopped or started</td>
</tr>
<tr>
<td>23</td>
<td>SERVER_PRINCIPAL_CHANGE_GROUP</td>
<td>This report will provide details regarding users created, deleted or modified. It will also include events regarding role and permission changes for users</td>
</tr>
<tr>
<td>24</td>
<td>DATABASE_CHANGE_GROUP</td>
<td>This report will provide details regarding database startup and shutdowns. It only includes events regarding status change of databases from offline or online</td>
</tr>
</tbody>
</table>

**Note:** Only available in Microsoft SQL Server 2012 or later.

Configured Server Audit Specification Properties pane is shown below:
4. Click **OK** to apply settings.

5. Right-click on earlier created **audit** and select **Enable**.

7. The above configuration generates event id “33205” for all configured audit specifications.

2.4 EventTracker agent configuration

2.4.1 Creating Event Filters

All the events generated by SQL through audit specifications are information events and are reported late. Thus, to aid the alerting of events in real-time event filters are to be configured.

Note: The log source matched in configurations below might change depending on the SQL instance name configured.
1. Logon to EventTracker manager workstation.

![EventTracker Control Panel](image1)

Figure 10

2. Open EventTracker control panel and click **EventTracker Agent Configuration**.

![EventTracker Agent Configuration](image2)

Figure 11
3. **Select Event Filters tab and click Filter Exception.**

Filter exception window opens.

![Filter Exception Window](Image)

4. Click **New**, and configure as shown below:

5. Event filter properties for **audit events** are shown below:

6. For **Single Instance**, Match in Source should be named as corresponding instance name. For **multiple instances**, leave it blank.

![Event Filter Details](Image)
7. Click **OK** to apply.
8. If events are enabled for **login success and failure**, create filters with configurations as shown below:
9. Event filter properties for **login success** event are shown below:

![Event Details](image)

**Figure 14**

10. Event filter properties for **login failure** event are shown below:
11. Click SAVE in agent configuration window to apply changes.

3. Extended events

Extended Events is a lightweight performance monitoring system that uses very few performance resources. It enables auditing for different actions, providing much granularity in the setup process and a wide coverage range of the SQL Server activity. **Below mentioned configuration must be applied to all the workstations, where SQL audit is required.**

3.1 Prerequisites

- **Microsoft SQL Server 2012 or later** must be installed.
- **Microsoft SQL Management Studio** for the respective version must be installed.
- **EventTracker agent 8 or later** must be installed on the SQL SERVER workstation.
- **PowerShell 5.1 or later** must be installed.
- **Administrative credentials** for script execution.

3.2 Creating extended event session

1. Contact Support for the **SQL extended events script pack** and download.
2. Extract downloaded zip file to the following path.

```
<EventTracker Installation Path>\MSSQL Server\n```

3. From extracted file location, double-click to open “SQL Extended Events.sql”.

![Figure 16](image)

4. In Microsoft SQL management studio, login with appropriate credentials.

5. In the SQL query, change the highlighted path to the desired location of the “.xel” file. Click Execute to create and implement an extended event session.

Preferred Location for “.xel” events:

```
<EventTracker Installation Path>\MSSQL Server\n```

Note: Create “MSSQL Server” folder if not present in the EventTracker Installation Path.
6. To view created session, navigate to **Object Explorer** > **Server Name** > **Management** > **Extended Events** > **Sessions** > **ObjectChange**.
7. The above configuration will create “.xel” file with all relevant audit events at the earlier mentioned location.

3.3 Parse extended event session log file

Xel files are readable only through SQL Management Studio. Thus, PowerShell is deployed for file format conversion and custom parsing in the interest of EventTracker.

1. From earlier mentioned extracted file location, find “EventTracker Task (SQL Extended Task).ps1”.

Figure 18

Figure 19
2. Logon to EventTracker Manager workstation with the administrative privileges.
3. Run the above “EventTracker Task (SQL Extended Task).ps1” program. It starts creating the task and logs will start sending to EventTracker.
4. Verify Task is created for SQL Extended events in TaskScheduler.

![Task Scheduler](image)

**Figure 20**

4. EventTracker Knowledge Pack

Once MSSQL Server is configured to send audit logs to EventTracker Manager, either through audit specifications or extended events. EventTracker will process the received logs and aid an administrator with informative reports, effective alerts, and dashboard visualizations.

4.1 Saved Searches

Extended Events:

**MSSQL Extended: Database created or deleted or modified**: This saved search will provide extended logs of the new database created and older databases altered or deleted.

**MSSQL Extended: Trigger created or deleted or modified**: This saved search will provide extended logs of the new database trigger created and older database triggers altered or deleted.
Integrate Microsoft SQL Server with EventTracker

**MSSQL Extended: Extended event session created or deleted or modified:** This saved search will provide extended logs of the new extended event session created and older extended event session altered or deleted.

**MSSQL Extended: Index created or deleted or modified:** This saved search will provide extended logs of the new database index created and older database index altered or deleted.

**MSSQL Extended: Schema created or deleted or modified:** This saved search will provide extended logs of the new database schema created and older database schema altered or deleted.

**MSSQL Extended: Stored procedure created or deleted or modified:** This saved search will provide extended logs of the new database stored procedure created and older database stored procedure altered or deleted.

**MSSQL Extended: Table created or deleted or modified:** This saved search will provide extended logs of the new database table created and older databases table altered or deleted.

**MSSQL Extended: User logon failure:** This saved search will provide user logon failure logs.

**MSSQL Extended: User logon success:** This saved search will provide user logon success logs.

**MSSQL Extended: View created or deleted or modified:** This saved search will provide extended logs of the new database view created and older database view altered or deleted.

**MSSQL Extended: Database backed up or restored:** This saved search will provide logs of database-backed up or restored.

**MSSQL Extended: Error details:** This saved search will provide SQL errors like syntax errors are generated.

**MSSQL Extended: Password reset or changed:** This saved search will provide extended logs of the user password reset or changed.

**MSSQL Extended: Permission granted or revoked or denied:** This saved search will provide extended logs of the user permissions granted or revoked or denied.

**MSSQL Extended: Role created or deleted or modified:** This saved search will provide extended logs of the new role created and older roles altered or deleted.

**MSSQL Extended: User created or deleted or modified:** This saved search will provide extended logs of the new database user-created and older database user altered or deleted.

**MSSQL Extended: User enabled or disabled or unlocked:** This saved search will provide extended logs of the new database user enabled, disabled, and unlocked.

**Note:** Following Saved Searches are created using **extended session** events. Similar Saved Searches are available through audit specifications as well, but Few saved searches support archive search only.
4.2 Alerts

MSSQL: Error detected - This alert is generated when SQL errors like syntax errors are generated.

MSSQL: Audit created deleted or modified - This alert is generated when audit, audit specification and extended event sessions are created, deleted or modified.

MSSQL: Database backed up or restored - This alert is generated when database backup is taken, or it is restored.

MSSQL: Database created or deleted or modified - This alert is generated when a new database is created, and older ones are deleted or modified.

MSSQL: Schema created or deleted or modified - This alert is generated when new database schema is created, and older ones are deleted or modified.

MSSQL: View created or deleted or modified - This alert is generated when the new database view is created, and older ones are deleted or modified.

MSSQL: User enabled or disabled or unlocked - This alert is generated when an existing login is enabled, disabled or unlocked.

MSSQL: Permission granted or revoked or denied - This alert is generated when permission is granted, revoked or denied to login or user.

MSSQL: Database and application role created or deleted or modified - This alert is generated when a new server or database role is created, and older ones are deleted or modified.

MSSQL: Stored procedure created or deleted or modified - This alert is generated when a new stored procedure is created, and older ones are deleted or modified.

MSSQL: Table created or deleted or modified - This alert is generated when a new table is created and older ones are deleted, truncated or modified.

MSSQL: Index created or deleted or modified - This alert is generated when the new table view is created, and older ones are deleted or modified.

MSSQL: Trigger created or deleted or modified - This alert is generated when a new table or database trigger is created, and older ones are deleted or modified.

MSSQL: User created or deleted or modified - This alert is generated when new login, user or credential is created, and older ones are deleted or modified.

MSSQL: User logon failure - This alert is generated when a user fails to login SQL server.
MSSQL: User and application role password reset or changed - This alert is generated when the password is changed or reset for login, credential or application role.

Note: Alerts are provided for both Audit and extended events.

4.3 Reports

The following reports are created using extended session events. Similar reports are available through audit specifications as well, but they will not have Host Name, Application Name and event category fields.

MSSQL Extended-Table created or deleted or modified - This report provides information related to table creation, deletion, and alteration.

Sample_Report:

<table>
<thead>
<tr>
<th>EventTime</th>
<th>Client Name</th>
<th>UserName</th>
<th>Client Application Name</th>
<th>Database Name</th>
<th>Instance Name</th>
<th>Event Category</th>
<th>Object Name</th>
<th>Query Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/02/16 04:19:30 PM</td>
<td>Contoso-Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>object_altered</td>
<td>Table_Employee</td>
<td>CREATE TABLE Employee_Temp_Audit ( Emp_ID INT, Emp_name varchar(100), Emp_Ssn decimal(10,2), Audit_Action varchar(100), Audit_Timestamp datetime )</td>
</tr>
<tr>
<td>08/02/16 04:19:30 PM</td>
<td>Contoso-Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>object_altered</td>
<td>Table_Employee</td>
<td>ALTER TABLE dbo.Table_Employee ADD Address nchar(10) NULL</td>
</tr>
<tr>
<td>08/02/16 04:19:30 PM</td>
<td>Contoso-Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>object_deleted</td>
<td>Table_Employee</td>
<td>DROP TABLE [dbo].[Table_2]</td>
</tr>
</tbody>
</table>

Log_Sample:

ENTRY:Timestamp : 08/02/16 04:19:30 PM
AppName : Microsoft SQL Server Management Studio
HostName : Contoso-Wkstn9
UserName : ADMIN\nick
EventDetails : object_altered
DataBase : master
Instance : MSSQLSERVER
ObjectType : USRTAB
ObjectName : Table_Employee
Statement : CREATE TABLE Employee_Temp_Audit ( Emp_ID INT, Emp_name varchar(100), Emp_Ssn decimal(10,2), Audit_Action varchar(100), Audit_Timestamp datetime )
SQLText : ALTER TABLE dbo.Table_Employee ADD Address nchar(10) NULL
Message : 
**MSSQL Extended** - Database created or deleted or modified - This report provides information related to database creation, deletion, and alteration.

Sample_Report:

<table>
<thead>
<tr>
<th>Event Time</th>
<th>Client Name</th>
<th>User Name</th>
<th>Application Name</th>
<th>Database Name</th>
<th>Event Category</th>
<th>Object Name</th>
<th>Query/Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>03/08/11 01:14:00 AM</td>
<td>Coreless</td>
<td>ADMIN\Nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>MSSQLSERVER</td>
<td>object_deleted</td>
<td>EmployeeDATA</td>
<td>DROP DATABASE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[EmployeeDATA]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[EmployeeDATA]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event Time</th>
<th>Client Name</th>
<th>User Name</th>
<th>Application Name</th>
<th>Database Name</th>
<th>Event Category</th>
<th>Object Name</th>
<th>Query/Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>03/08/11 01:13:28 AM</td>
<td>Coreless</td>
<td>ADMIN\Nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>MSSQLSERVER</td>
<td>object_created</td>
<td>EmployeeDATA</td>
<td>CREATE DATABASE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[EmployeeDATA]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[EmployeeDATA]</td>
</tr>
</tbody>
</table>
Log_sample:

ENTRY: Timestamp : 08/02/16 01:14:00 PM
AppName : Microsoft SQL Server Management Studio
HostName : Contoso-Wkstn9
UserName : ADMIN\nick
EventDetails : object_deleted
DataBase : master
Instance : MSSQLSERVER
ObjectType : DATABASE
ObjectName : EmployeeDATA
Statement :

SQLText : /***** Object: Database [EmployeeDATA]
Script Date: 08/02/16 01:13:55 PM ******/
DROP DATABASE [EmployeeDATA]
Message :
FILE:d:\SQL\sql logs.txt
TYPE: MULTILINE
FIELD: *

MSSQL Extended-View created or deleted or modified - This report provides information related to database view creation, deletion and alteration.

Sample_Report:

Log_Sample:

ENTRY: Timestamp : 08/02/16 04:22:10 PM
AppName : Microsoft SQL Server Management Studio
HostName : Contoso-Wkstn9
UserName : ADMIN\nick

MSSQL Extended-View created or deleted or modified - This report provides information related to database view creation, deletion and alteration.

Sample_Report:

Log_Sample:
**Integrate Microsoft SQL Server with EventTracker**

**EventDetails:** object_deleted  
**DataBase:** master  
**Instance:** MSSQLSERVER  
**ObjectType:** VIEW  
**ObjectName:** NewView  
**Statement:**  
**SQLText:**  
```sql  
/****** Object: View [dbo].[NewView]  
Script Date: 08/02/16 04:22:05 PM ******/  
DROP VIEW [dbo].[NewView]  
```

**Message:**  
FILE:d:\SQL\sql logs.txt  
**TYPE:** MULTILINE  
**FIELD:** *

**MSSQL Extended Stored procedure created or deleted or modified** - This report provides information related to stored procedure creation, deletion and alteration.

**Sample Report:**

<table>
<thead>
<tr>
<th>Event Time</th>
<th>Client Name</th>
<th>UserName</th>
<th>Client Application Name</th>
<th>Database Name</th>
<th>Instance</th>
<th>Event Category</th>
<th>Object Name</th>
<th>Query Text</th>
</tr>
</thead>
</table>
| 08/02/16 04:18:32 PM | Contoso-Wkstn9 | ADMIN\nick | Microsoft SQL Server Management Studio | master | MSSQLSERVER | object_deleted | NewProcedure | CREATE PROCEDURE NewProcedure  

```sql  
AS  
BEGIN  
-- SET NOCOUNT ON  
-- to prevent extra result sets from  
-- interfering with  
-- SELECT statements.  
SET NOCOUNT ON,  
BEGIN  
SELECT au.[au_name], au.[au_title]  
FROM au  
INNER JOIN author ON au.[author_id] = au.[author_id]  
RETURN  
END  
END  ```

<table>
<thead>
<tr>
<th>Event Time</th>
<th>Client Name</th>
<th>UserName</th>
<th>Client Application Name</th>
<th>Database Name</th>
<th>Instance</th>
<th>Event Category</th>
<th>Object Name</th>
<th>Query Text</th>
</tr>
</thead>
</table>
| 08/02/16 04:17:47 PM | Contoso-Wkstn9 | ADMIN\nick | Microsoft SQL Server Management Studio | master | MSSQLSERVER | object_created | NewProcedure | CREATE PROCEDURE NewProcedure  

```sql  
AS  
BEGIN  
-- SET NOCOUNT ON  
-- to prevent extra result sets from  
-- interfering with  
-- SELECT statements.  
SET NOCOUNT ON,  
BEGIN  
RETURN  
END  ```

**Log Sample:**

**ENTRY:** Timestamp: 08/02/16 04:18:32 PM  
AppName: Microsoft SQL Server Management Studio  
HostName: Contoso-Wkstn9  
UserName: ADMIN\nick  
EventDetails: object_deleted  
DataBase: master
Instance : MSSQLSERVER  
ObjectType : PROC  
ObjectName : NewProcedure  
Statement :  
```
SQLText
/
****** Object: StoredProcedure  
[dbo].[Author_Title]  
Script Date: 08/02/16 04:18:27 PM ******
DROP PROCEDURE [dbo].[NewProcedure]
```

Message :  
FILE:d:\SQL\sql logs.txt  
TYPE:MULTILINE  
FIELD: *

### MSSQL Extended-Index created or deleted or modified
This report provides information related to table index creation, deletion and alteration.

#### Sample_Report:

<table>
<thead>
<tr>
<th>EventTime</th>
<th>Client Name</th>
<th>UserName</th>
<th>Client Application Name</th>
<th>Database Name</th>
<th>Instance Name</th>
<th>Event Category</th>
<th>Object Name</th>
<th>Query Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/02/16</td>
<td>Contoso-Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>object_created</td>
<td>ClusteredIndex</td>
<td>CREATE UNIQUE CLUSTERED INDEX [dbo].[Table_1] ON [dbo].[Table_1]</td>
</tr>
</tbody>
</table>
|             |                 |               |                          |               |               |                |               | { [mm.m] ASC,  
|             |                 |               |                          |               |               |                | [nnn.n] ASC,  
|             |                 |               |                          |               |               |                | [PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
|             |                 |               |                          |               |               |                | IGNORE_DUP_KEY = OFF,  
|             |                 |               |                          |               |               |                | DROP_EXISTING = OFF, ONLINE =  
|             |                 |               |                          |               |               |                | ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) |

| 08/02/16    | Contoso-Wkstn9  | ADMIN\nick    | Microsoft SQL Server Management Studio | master        | MSSQLSERVER   | object_altered | ClusteredIndex  | DROP INDEX [ClusteredIndex] ON [dbo].[Table_1] WITH ( ONLINE = OFF ) |

#### Log_Sample:

ENTRY:Timestamp : 08/02/16 04:15:57 PM  
AppName : Microsoft SQL Server Management Studio  
HostName : Contoso-Wkstn9  
UserName : ADMIN\nick  
EventDetails : object_altered  
DataBase : master  
Instance : MSSQLSERVER
Integrate Microsoft SQL Server with EventTracker

ObjectType : INDEX
ObjectName   : ClusteredIndex-20160714-164229
Statement    :
SQLText      : ****** Object: Index [ClusteredIndex-]
Script Date: 08/02/16 04:15:52 PM ******/
DROP INDEX [ClusteredIndex-] ON [dbo].[Table_1] WITH ( ONLINE = OFF )
Message      :
FILE:d:\SQL\sql logs.txt
TYPE:MULTILINE
FIELD: *

MSSQL Extended-Trigger created or deleted or modified - This report provides information related to table and database trigger creation, deletion and alteration.

Sample_Report:

<table>
<thead>
<tr>
<th>EventTime</th>
<th>Client Name</th>
<th>User Name</th>
<th>Client Application Name</th>
<th>Database Name</th>
<th>Instance Name</th>
<th>Event Category</th>
<th>Object Name</th>
<th>Query Text</th>
</tr>
</thead>
</table>
| 08/02/16    | Contoso-Wkstn9 | ADMIN\nick | Microsoft SQL Server Management Studio | MSSQLSERVER | object_created | trgAfterInsert | [dbo].Employee_Test | CREATE TRIGGER trgAfterInsert ON [dbo].Employee_Test
FOR INSERT
AS
declare @empid int;
declare @empname varchar(100);
declare @empsal decimal(10,2);
declare @audit_action varchar(100);

select @empid=Emp_Id from inserted ;
select @empname=Emp_Name from inserted ;
select @empsal=Emp_Sal from inserted ;
set @audit_action=Inserted Record -- After Insert Trigger ;
insert into |

Log_Smaple:

ENTRY:Timestamp : 08/02/16 04:20:10 PM
AppName       : Microsoft SQL Server Management Studio
HostName      : Contoso-Wkstn9
UserName      : ADMIN\nick
EventDetails  : object_created
DataBase      : master
Integrate Microsoft SQL Server with EventTracker

Instance     : MSSQLSERVER
ObjectType   : TRIGGER
ObjectName   : trgAfterInsert
Statement    :
SQLText      : CREATE TRIGGER trgAfterInsert
ON [dbo].[Employee_Test] 
    FOR INSERT
    AS
        declare @empid int;
        declare @empname varchar(100);
        declare @empsal decimal(10,2);
        declare @audit_action varchar(100);

        select @empid=i.Emp_ID from inserted i;
        select @empname=i.Emp_Name from inserted i;
        select @empsal=i.Emp_Sal from inserted i;
        set @audit_action='Inserted Record -- After Insert Trigger.,'

        insert into Employee_Test_Audit
        (Emp_ID,Emp_Name,Emp_Sal,Audit_Action,Audit_Timestamp)
        values(@empid,@empname,@empsal,@audit_action,getdate())

Message      :
FILE:d:\sql logs.txt
TYPE:MULTILINE
FIELD: *

**MSSQL Extended-User created or deleted or modified** - This report provides information related to login, user and credential creation, deletion and alteration.

**Sample_Report:**

<table>
<thead>
<tr>
<th>EventTime</th>
<th>Client Name</th>
<th>User Name</th>
<th>Client Application Name</th>
<th>Database Name</th>
<th>Instance Name</th>
<th>Query Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/02/16 04:16:05 PM</td>
<td>Contoso- Wkstrn9</td>
<td>ADMIN nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>-- ALTER LOGIN</td>
</tr>
<tr>
<td>08/02/16 04:20:47 PM</td>
<td>Contoso- Wkstrn9</td>
<td>ADMIN nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>-- CREATE LOGIN</td>
</tr>
<tr>
<td>08/02/16 04:20:56 PM</td>
<td>Contoso- Wkstrn9</td>
<td>ADMIN nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>DROP LOGIN [Riley]</td>
</tr>
</tbody>
</table>

**Log_Sample:**
**MSSQL Extended-Database and application role created or deleted or modified** - This report provides information related to the server, database and application role creation, deletion and alteration.

Sample_Report:

<table>
<thead>
<tr>
<th>Event Time</th>
<th>Client Name</th>
<th>User Name</th>
<th>Client Application</th>
<th>Database Name</th>
<th>Instance Name</th>
<th>Query Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/02/16 04:16:05 PM Wkstn9</td>
<td>Contoso-Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>-- ALTER LOGIN</td>
</tr>
<tr>
<td>08/02/16 04:20:47 PM Wkstn9</td>
<td>Contoso-Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>-- CREATE LOGIN</td>
</tr>
<tr>
<td>08/02/16 04:20:58 PM Wkstn9</td>
<td>Contoso-Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>DROP LOGIN [File]</td>
</tr>
</tbody>
</table>

Log_Sample:
Integrate Microsoft SQL Server with EventTracker

ENTRY:Timestamp : 08/02/16 04:16:42 PM
AppName : Microsoft SQL Server Management Studio
HostName : Contoso-Wkstn9
UserName : ADMIN\nick
EventDetails : sql_statement_completed
DataBase : master
Instance : MSSQLSERVER
ObjectType : 
ObjectName : Statement : CREATE ROLE [New Role] AUTHORIZATION [db_backupoperator]
Message : FILE:d:\SQL\sql logs.txt
TYPE: MULTILINE
FIELD: *

**MSSQL Extended-Schema created or deleted or modified** - This report provides information related to database schema creation, deletion and alteration.

Sample_Report:

<table>
<thead>
<tr>
<th>EventTime</th>
<th>Client Name</th>
<th>User Name</th>
<th>Client Application Name</th>
<th>Database Name</th>
<th>Instance</th>
<th>Event Category</th>
<th>Object Name</th>
<th>Query Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/02/16 04:16:42 PM</td>
<td>Contoso-Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>object_created</td>
<td>NewSchema</td>
<td>CREATE SCHEMA [NewSchema] AUTHORIZATION [db_backupoperator]</td>
</tr>
<tr>
<td>08/02/16 04:17:34 PM</td>
<td>Contoso-Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>object_deleted</td>
<td>NewSchema</td>
<td>DROP SCHEMA [NewSchema]</td>
</tr>
</tbody>
</table>

Log_Sample:

ENTRY:Timestamp : 08/02/16 04:17:34 PM
AppName : Microsoft SQL Server Management Studio
HostName : Contoso-Wkstn9
UserName : ADMIN\nick
EventDetails : object_deleted
DataBase : master
Instance : MSSQLSERVER
ObjectType : SCHEMA
ObjectName : NewSchema
Statement :
Integrate Microsoft SQL Server with EventTracker

MSSQL Extended-User and application role password reset or changed - This report provides information related to login, credential and application role’s creation, deletion and alteration.

Sample Report:

<table>
<thead>
<tr>
<th>EventTime</th>
<th>Client Name</th>
<th>User Name</th>
<th>Client Application Name</th>
<th>Database Name</th>
<th>Instance Name</th>
<th>Query Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/02/16 04:17:29 PM</td>
<td>Contoso-Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>-- ALTER LOGIN</td>
</tr>
</tbody>
</table>

Log Sample:

ENTRY:Timestamp : 08/02/16 04:16:05 PM
AppName : Microsoft SQL Server Management Studio
HostName : Contoso-Wkstn9
UserName : ADMIN\nick
EventDetails : sql_statement_completed
DataBase : master
Instance : MSSQLSERVER
ObjectType : 
ObjectName : 
Statement : -- ALTER LOGIN
SQLText : *password--------------------------------------
Message : 
FILE:d:\SQL\sql logs.txt
TYPE:MULTILINE
FIELD: *

MSSQL Extended-User enabled or disabled or unlocked - This report provides information related to login account enabled, disabled and unlocked.
Sample_Report:

<table>
<thead>
<tr>
<th>Event Time</th>
<th>Client Name</th>
<th>User Name</th>
<th>Client Application Name</th>
<th>Database Name</th>
<th>Instance Name</th>
<th>Query Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/02/16 06:04:49 PM</td>
<td>Contoso- Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>ALTER LOGIN [Rick] DISABLE</td>
</tr>
<tr>
<td>08/02/16 06:05:10 PM</td>
<td>Contoso- Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>ALTER LOGIN [Riley] ENABLE</td>
</tr>
</tbody>
</table>

Log_Sample:

```
ENTRY:Timestamp : 08/02/16 06:04:49 PM
AppName : Microsoft SQL Server Management Studio
HostName : Contoso-Wkstn9
UserName : ADMIN\nick
EventDetails : sql_statement_completed
DataBase : master
Instance : MSSQLSERVER
ObjectType :
ObjectName :
Statement : ALTER LOGIN [Rick] DISABLE
SQLText : ALTER LOGIN [Rick] DISABLE
Message :
FILE: d:\SQL\sql logs.txt
TYPE: MULTILINE
FIELD: *
```

**MSSQL Extended-Database backed up or restored** - This report provides information related to database backup and restore.

Sample_Report:
Integrate Microsoft SQL Server with EventTracker

Log_Sample:

<table>
<thead>
<tr>
<th>Event Time</th>
<th>Client Name</th>
<th>User Name</th>
<th>Client Application Name</th>
<th>Database Name</th>
<th>Instance Name</th>
<th>Query Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/02/16 12:12:30 PM</td>
<td>Contoso-Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>BACKUP DATABASE [EmployeeDB] TO DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Backup\EmployeeDB_Log Backup\TEMP\Full Database Backup', SKIP, NORECOVERY, NOUNLOAD, STATS = 10</td>
</tr>
<tr>
<td>08/02/16 12:13:03 PM</td>
<td>Contoso-Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>RESTORE DATABASE [EmployeeDB] FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Backup\EmployeeDB.bak' WITH FILE = 2, NOUNLOAD, STATS = 5</td>
</tr>
</tbody>
</table>

MSSQL Extended - Permission granted or revoked or denied - This report provides information related to permission granted, revoked and denied to a user or login.
MSSQL Extended-Extended event session created or deleted or modified* - This report provides information related to extended event session creation, deletion, and alteration.

**Sample_Report:**

<table>
<thead>
<tr>
<th>EventTime</th>
<th>Client Name</th>
<th>UserName</th>
<th>Client Application Name</th>
<th>Database Name</th>
<th>Instance Name</th>
<th>Query Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/02/16 04:16:21 PM</td>
<td>Contoso-Wktn9</td>
<td>ADMINnick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>GRANT INSERT ON [dbo].[Table_1] TO [Rick] WITH GRANT OPTION</td>
</tr>
<tr>
<td>08/02/16 04:16:33 PM</td>
<td>Contoso-Wktn9</td>
<td>ADMINnick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>master</td>
<td>MSSQLSERVER</td>
<td>DENY ALTER ANY CREDENTIAL TO [Rick]</td>
</tr>
</tbody>
</table>

**Log_Sample:**

```
ENTRY:Timestamp : 08/02/16 04:16:21 PM
AppName : Microsoft SQL Server Management Studio
HostName : Contoso-Wktn9
UserName : ADMIN
nick
EventDetails : sql_statement_completed
DataBase : master
Instance : MSSQLSERVER
ObjectType :
ObjectName :
Statement : GRANT INSERT ON [dbo].[Table_1] TO [Rick] WITH GRANT OPTION
SQLText : GRANT INSERT ON [dbo].[Table_1] TO [Rick] WITH GRANT OPTION
Message :
FILE:d:\SQL\sql logs.txt
TYPE: MULTILINE
FIELD: *
```
Integrate Microsoft SQL Server with EventTracker

Log_Sample:

ENTRY:Timestamp : 08/02/16 04:14:55 PM
AppName : Microsoft SQL Server Management Studio
HostName : Contoso-Wkstn9
UserName : ADMIN\nick
EventDetails : object_created
DataBase : master
Instance : MSSQLSERVER
ObjectType : SRVXESES
ObjectName : ObjectChange
Statement :
SQLText : CREATE EVENT SESSION [Test] ON SERVER
ADD EVENT sqlserver.assembly_load
WITH (STARTUP_STATE=OFF)
Message :
FILE:d:\SQL\sql logs.txt
TYPE:MULTILINE
FIELD: *

MSSQL Extended-User logon success - This report provides information related to user logon success.

Sample_Report:

<table>
<thead>
<tr>
<th>Event Time</th>
<th>Client Name</th>
<th>Client Address</th>
<th>User Name</th>
<th>Client Application Name</th>
<th>Authentication Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/02/16 01:51:11 AM</td>
<td>Contoso-Wkstn5</td>
<td>192.168.1.25</td>
<td>ADMIN\nick</td>
<td>Microsoft PowerShell</td>
<td>Windows authentication</td>
</tr>
<tr>
<td>08/02/16 05:50:13 PM</td>
<td>Contoso-Wkstn5</td>
<td>&lt;local machine&gt;</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>SQL authentication</td>
</tr>
</tbody>
</table>

Log_Sample:

ENTRY:Timestamp : 08/02/16 01:50:01 PM
AppName : Microsoft SQL Server Management Studio
HostName : Contoso-Wkstn5
UserName : ADMIN\nick
EventDetails : error_reported
DataBase :
Instance :
ObjectType :
ObjectName :
Statement :
SQLText :
Message: Login succeeded for user 'ADMIN\nick'. Connection made using Windows authentication. [CLIENT: 192.168.1.25]

**MSSQL Extended-Error details** - This report provides information related to errors generated by SQL.

Sample_Report:

<table>
<thead>
<tr>
<th>Event Time</th>
<th>Client Name</th>
<th>UserName</th>
<th>Client Application Name</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/02/16 04:14:41 PM</td>
<td>Contoso-Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>Invalid object name 'master.dbo.set_values'.</td>
</tr>
<tr>
<td>08/02/16 04:14:46 PM</td>
<td>Contoso-Wkstn9</td>
<td>ADMIN\nick</td>
<td>Microsoft SQL Server Management Studio</td>
<td>Incorrect syntax near '&lt;'.</td>
</tr>
</tbody>
</table>

Log_Sample:

```plaintext
ENTRY:Timestamp      : 08/02/16 04:14:46 PM
AppName              : Microsoft SQL Server Management Studio
HostName             : Contoso-Wkstn9
UserName             : ADMIN\nick
EventDetails         : error_reported
DataBase             : ...
Instance             : ...
ObjectType           : ...
ObjectName           : ...
Statement            : ...
SQLText              : ...
Message              : Incorrect syntax near '<'.
FILE:d:\SQL\sql logs.txt
TYPE:Multiline
FIELD: *             
```

**MSSQL Extended-User logon failure** - This report provides information related to a user logon failure.

Sample_Report:

<table>
<thead>
<tr>
<th>Event Time</th>
<th>Client Name</th>
<th>Client Application Name</th>
<th>UserName</th>
<th>Client Address</th>
<th>Failure Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/01/16 06:10:36 PM</td>
<td>Contoso-Wkstn5</td>
<td>Microsoft SQL Server Management Studio</td>
<td>Rick</td>
<td>&lt;local machine&gt;</td>
<td>An attempt to login using SQL authentication failed. Server is configured for Windows authentication only</td>
</tr>
</tbody>
</table>
Integrate Microsoft SQL Server with EventTracker

Log_Sample:

```
Timestamp : 08/01/16 06:10:36 PM
AppName   : Microsoft SQL Server Management Studio
HostName  : Contoso-Wkstn5
UserName  : ADMIN\rick
EventDetails : error_reported
DataBase   :
Instance   :
ObjectType  :
ObjectName  :
Statement   :
SQLText     :
Message     : Login failed for user ‘Rick’. An attempt to login using SQL authentication failed. Server is configured for Windows authentication only. [CLIENT: <local machine>]
```

4.4 Dashboards

The following dashboards are created using Extended session events.

**MSSQL– Logon success**: This dashboard can view SQL server user login success usernames.

![MSSQL - Logon success](image)

*Figure 21*
**MSSQL – Logon failed:** This dashboard can view SQL server user login failed usernames.

![MSSQL Logon failed dashboard](image1)

**Figure 22**

**MSSQL Extended – Object management activities:** This dashboard can show DDL activities created, altered and deleted for database view, trigger, procedure, index, and schema.

![MSSQL Extended - Object management activities](image2)

**Figure 23**
**MSSQL Extended – Database management:** This dashboard can view new database created and older ones are deleted or modified.

![Figure 24](image1.png)

**MSSQL Extended – Extended event session management:** This dashboard can view extended event session created, altered, and deleted.

![Figure 25](image2.png)
MSSQL Extended – Table management: This dashboard can view a database table created, altered, and deleted.

![MSSQL Extended - Table management](image)

**Figure 26**

**Note:** Only available if extended events are enabled.

5. Importing knowledge pack into EventTracker

**NOTE:** Import knowledge pack items in the following sequence:

- Categories
- Alerts
- Token Template/ Parsing Rules
- Flex Reports
- Knowledge Objects
- Dashboards

1. Launch the EventTracker Control Panel.
2. Double click Export-Import Utility.
3. Click the **Import** tab.

### 5.1 Categories

1. Once you have opened “Export-Import Utility” via “EventTracker Control Panel”, click the **Category** option, and then click browse.

2. Navigate to the knowledge pack folder and select the file with the extension “.iscat”, like “Categories_MSSQL.iscat” and then click “Import”.

![Figure 27](image1.png)

![Figure 28](image2.png)
Integrate Microsoft SQL Server with EventTracker

EventTracker displays a success message:

5.2 Alerts

1. Once you have opened “Export-Import Utility” via “EventTracker Control Panel”, click Alert option, and then click browse ...
2. Navigate to the knowledge pack folder and select the file with the extension “.isalt”, e.g. “Alerts_MSSQL.isalt” and then click “Import”.

Figure 29

Figure 30
Integrate Microsoft SQL Server with EventTracker

Figure 31

EventTracker displays a success message:

Figure 32

5.3 Token Templates

For importing “Token Template”, navigate to the EventTracker manager web interface.

1. Click Parsing Rules under the Admin option in the EventTracker manager web interface.
Next, click “Template” tab and then click “Import Configuration”.

2. Now, click the “Browse” and navigate to the knowledge packs folder (type C:\Program Files (x86)\Prism Microsystems\EventTracker\Knowledge Packs” in the navigation bar) where “.ettd”, e.g. “Templates_MSSQL.ettd” file is located. Wait for a few seconds, as templates will be loaded. Once you see the templates, click desired templates and click “Import”:
5.4 Flex Reports

1. In the EventTracker control panel, select “Export/Import utility” and select the “Import tab”. Then, click Reports option, and choose “New (*.etcrx)”: 

2. Once you have selected “New (*.etcrx)”, a new pop-up window will appear. Click the “Select File” button and navigate to the knowledge pack folder and select file with the extension “.etcrx”, e.g. “Reports_MSSQL.etcrx”.
3. Wait while reports are being populated in the below tables. Now, select all the relevant reports and then click **Import** button.

EventTracker displays a success message:

5.5 **Knowledge Objects**

1. Click **Knowledge objects** under the **Admin** option in the EventTracker web interface.
2. Next, click “import object”:

3. A pop-up box will appear, click “Browse” in that and navigate to the knowledge packs folder (type “C:\Program Files (x86)\Prism Microsystems\EventTracker\Knowledge Packs” in the navigation bar) with the extension “.etko”, e.g. “KO_MSSQL.etko” and then click the “Upload” button.

4. Wait while EventTracker populates all the relevant knowledge objects. Once the objects are displayed, select the required ones and click on the “Import” button:
5.6 Dashboards

1. Login to the EventTracker web interface.
2. Navigate to Dashboard → My Dashboard.
3. In “My Dashboard”, click Import:

![Figure 45](image)

4. Select browse and navigate to the knowledge pack folder (type “C:\Program Files (x86)\Prism Microsystems\EventTracker\Knowledge Packs” in the navigation bar) where “.etwd”, e.g. “Dashboard_MSSQL.etwd” is saved and click on “Upload” button.

![Figure 46](image)

5. Wait while EventTracker populates all the available dashboards. Now, choose “Select All” and click “Import”.

---

Integrate Microsoft SQL Server with EventTracker
6. Verifying knowledge pack in EventTracker

6.1 Categories

1. Login to the EventTracker web interface.
2. Click Admin dropdown, and then click Categories.
3. In Category Tree to view imported categories, click “Search” and search with the “MSSQL”. You will see the below results.
6.2 Alerts

1. In the EventTracker web interface, click the Admin dropdown, and then click Alerts.
2. In the search box enter “MSSQL” and then click Search.
   EventTracker displays an alert related to MSSQL:

Figure 50

6.3 Token Templates

1. In the EventTracker web interface, click Admin, and then click “Parsing Rules”.
2. In the “Template” tab, click on the “MSSQL” group folder to view the imported Token.

Figure 51

6.4 Flex Reports

1. In the EventTracker web interface, click Reports menu, and then select the Report Configuration.
2. In **Reports Configuration** pane, select the **Defined** option.
3. Click on the “**MSSQL**” group folder to view the imported reports.

![Image of Report Configuration]

**Figure 53**

### 6.5 Knowledge Objects

1. In the **EventTracker** web interface, click the **Admin** dropdown, and then click **Knowledge Objects**.
2. In the **Knowledge Object** tree, expand the “**MSSQL**” group folder to view the imported Knowledge objects.

![Image of Knowledge Objects]

**Figure 54**

### 6.6 Dashboards

1. In the **EventTracker** web interface, Click Home and select “**My Dashboard**”.
2. In “MSSQL” dashboard you should be now able to see something like this: