SQL Event Analyzer

Indhold

[1. Introduktion 3](#_Toc465883146)

[2. Rettigheder 8](#_Toc465883147)

[3. Standard kolonner 9](#_Toc465883148)

[4. Brugerdefinerede kolonner 10](#_Toc465883149)

[4.1 Generering af output visning 12](#_Toc465883150)

[4.2 Input- og output typer 13](#_Toc465883151)

[4.3 CLR understøttelse 15](#_Toc465883152)

[4.4 Arbejde med brugerdefinerede kolonner 16](#_Toc465883153)

[4.5 CLR funktionalitet sammen med SQL 20](#_Toc465883154)

[4.6 Renset TextData 22](#_Toc465883155)

[4.7 Eksempel på oprettelse af index på en brugerdefineret kolonne 23](#_Toc465883156)

[4.8 Abonnement 25](#_Toc465883157)

[5. Hændelser fra tracefiler 27](#_Toc465883158)

[6. Realtime hændelser 27](#_Toc465883159)

[6.1 Ændring af tracing funktionalitet 28](#_Toc465883160)

[7. Søgning og sortering 29](#_Toc465883161)

[8. Filtre 30](#_Toc465883162)

[8.1 Filter 1 32](#_Toc465883163)

[8.2 Filter 2 33](#_Toc465883164)

[8.3 Eksempel på filtrering 35](#_Toc465883165)

[9. Kørsel i uovervåget tilstand 36](#_Toc465883166)

[9.1 Autogenerering af aggregeret statistik 41](#_Toc465883167)

[9.2 Kørsel i service context 43](#_Toc465883168)

[9.3 Returkoder fra uovervåget kørsel 44](#_Toc465883169)

[9.4 Send kørselslog til web service 44](#_Toc465883170)

[10. Kørsel i optagetilstand 44](#_Toc465883171)

[11. Statistik 45](#_Toc465883172)

[11.1 Filtreret statistik 49](#_Toc465883173)

[11.2 Performanceforbedring ved arbejde med statistik 49](#_Toc465883174)

[12. Tidslinie 49](#_Toc465883175)

[13. Parametre 51](#_Toc465883176)

## Introduktion

SQL Event Analyzer muliggør visning og håndtering af hændelser på en SQL Server.

Der kan enten arbejdes med hændelser gemt i tracefiler eller realtime hændelser.



Når Tracefiler og realtime hændelser importeres, bliver de importeret til en tabel på SQL Serveren (fremover refereret til som en "TraceData tabel"). En TraceData tabel er navngivet i følgende format:

TraceData\_[Session Id]

Hvor Session Id er en guid.

Eksempelvis: TraceData\_f63631c1-5824-4dce-b631-f45e187c3cc1

Data gemt i en TraceData tabel vil blive refereret til som en "session". En session vil derfor kunne indeholde en eller flere importerede tracefiler eller realtime hændelser.

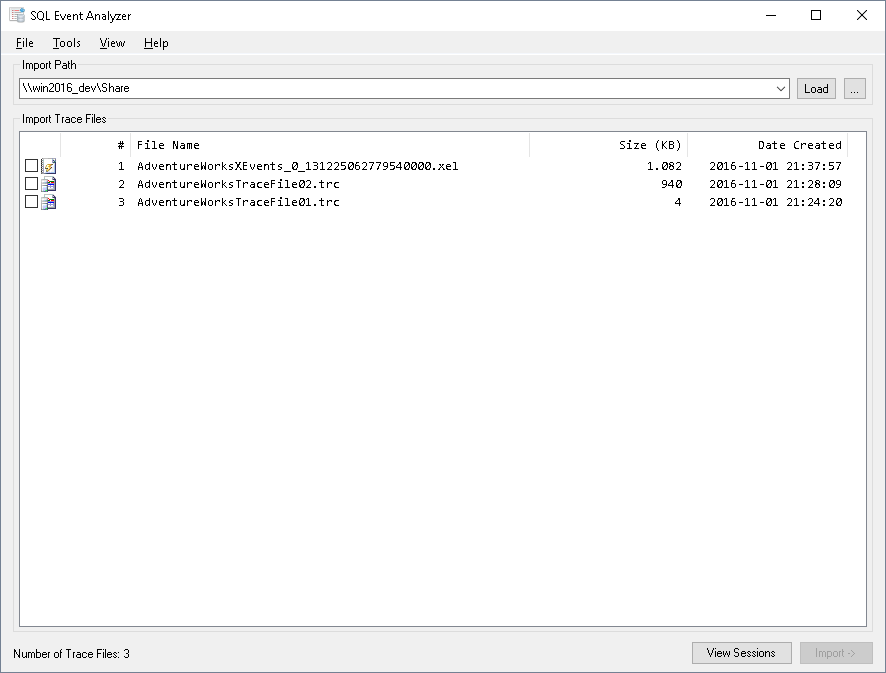
Med en TraceData tabel som input, præsenteres hændelserne som output på listeform opdelt i kolonner. Output kolonner består af en eller flere standard kolonner eventuelt kombineret med brugerdefinerede kolonner.

Der understøttes tracefiler i følgende formater:

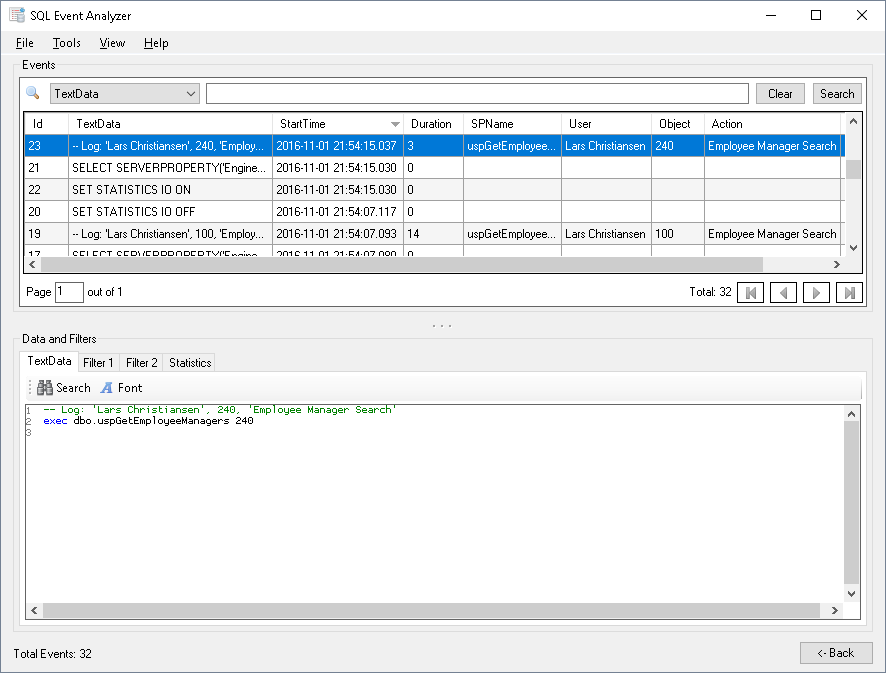
* SQL Server Profiler
* Extended Events

Bemærk, at Extended Events ikke er understøttet på SQL Server versioner ældre end 2012.

Eksempel på input fra tracefiler:



Eksempel på output visning:

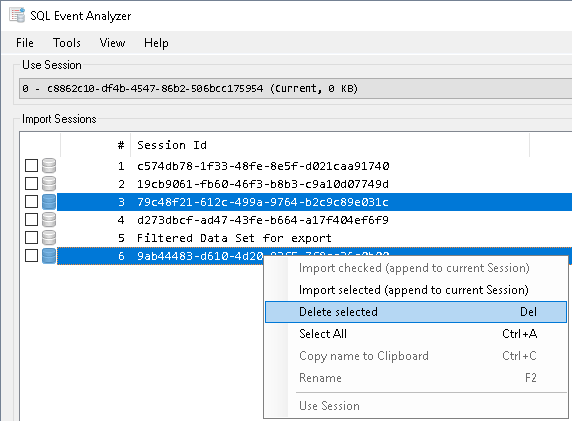


Ved første kørsel af SQL Event Analyzer, vil en database på SQL Serveren med navnet "SQLEventAnalyzer" blive oprettet. Hvis databasen findes i forvejen, vil denne database blive benyttet.

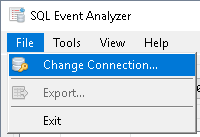
I SQLEventAnalyzer databasen håndteres TraceData tabellerne.

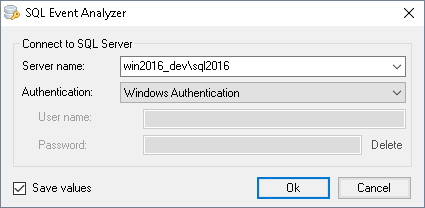
Når forbindelsen til SQL Serveren lukkes, kan det vælges at TraceData tabellerne automatisk slettes. Som standard vil TraceData tabellerne ikke blive slettet, så man efterfølgende kan forbinde til en session med allerede importeret data.

Gemte sessioner kan manuelt slettes fra session listen:

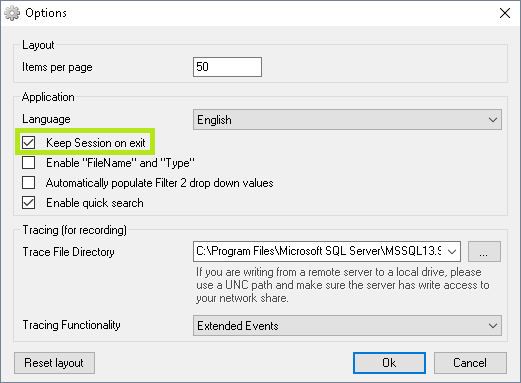


Hvis SQL Event Analyzer er opsat til at slette sessioner ved afslutning, kan dette ignoreres ved at holde Shift tasten nede når SQL Event Analyzer lukkes, eller ved at holde Shift tasten nede mens der vælges "Ok" når der oprettes en ny forbindelse til en SQL Server via "Skift forbindelse...":

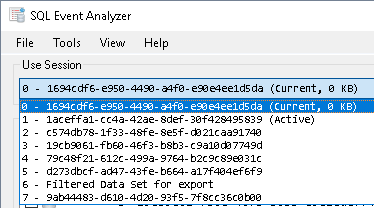




Om sessioner skal beholdes ved afslutning kan angives i "Indstillinger" menuen:



Hvis en anden bruger benytter en session, vil sessionen fremstå som "aktiv" i dropdown listen:



## Rettigheder

Den bruger, der forbinder til SQL Serveren, hvor SQL Event Analyzer skal arbejde, skal have følgende rettighed sat:

* "db\_owner" rettighed til "SQLEventAnalyzer" databasen
* "Alter any connection" rettighed på server niveau

Hvis SQL Server 2008 R2 eller ældre benyttes, skal brugeren have følgende rettighed:

* "Alter trace" rettighed på server niveau

Hvis SQL Server 2012 eller nyere benyttes, skal brugeren have følgende rettigheder:

* "Alter trace" rettighed på server niveau
* "View server state" rettighed på server niveau

Se afsnittet "Ændring af tracing funktionalitet" for mere information om tracing funktionalitet.

Bemærk, hvis den brugerkonto, der benytter SQL Event Analyzer, ikke har lokal administrator rettigheder, vil følgende ikke kunne gemmes:

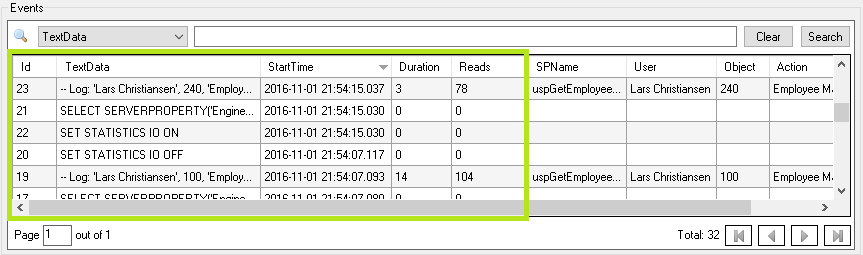
* Indstillinger
* Gemte søgninger
* Ændringer i brugergrænsefladen

## Standard kolonner

Hændelser indeholder som standard altid følgende data:

* Id: Autogenereret fortløbende Id nummer
* TextData: Indeholder den eksekverede SQL forespørgsel
* FileName: Den pågældende tracefil, hvor hændelsen eksisterer
* Type: Angiver hændelsestypen
* SPID: Det tilknyttede SPID
* Duration: Hændelsens eksekveringstid i millisekunder
* StartTime: Hændelsens starttidspunkt
* Reads: Antal reads for hændelsen
* Writes: Antal writes for hændelsen
* CPU: Hændelsens samlede CPU eksekveringstid i millisekunder
* Rows: Antal rækker, der er er arbejdet med i hændelsen

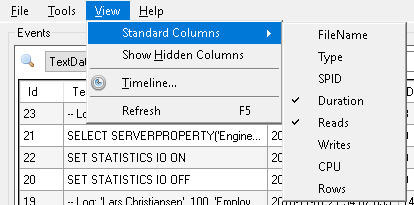
Ovenstående data vil i output visningen fremgå som standard kolonner:



Bemærk, at der som standard ikke vises følgende kolonner:

* FileName
* Type
* SPID
* Writes
* CPU
* Rows

Visning af standard kolonner kan vælges i "Vis", "Standard kolonner":



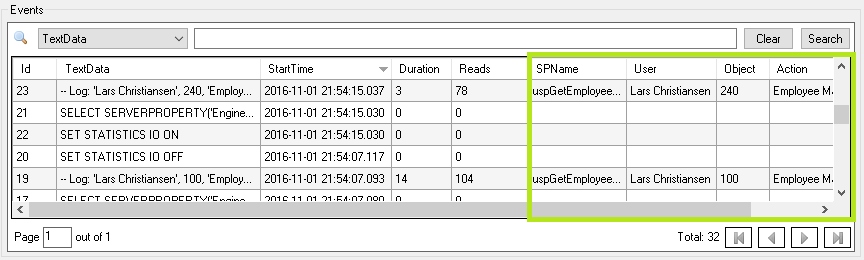
Det er ikke muligt at fravælge visning af "Id", "TextData" og "StartTime" kolonnerne.

En standard kolonne kan tilgåes fra andre steder i SQL Event Analyzer udfra dens navn. F.eks. er det muligt at tilgå en standard kolonne i en brugerdefineret kolonne.

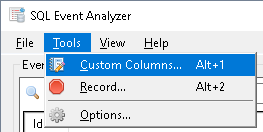
## Brugerdefinerede kolonner

Det er muligt at tilknytte ekstra brugerdefinerede kolonner ud over standard kolonnerne.

Brugerdefinerede kolonner vises i forlængelse af standard kolonnerne:



Brugerdefinerede kolonner tilgåes fra "Værktøjer", "Brugerdefinerede kolonner...":

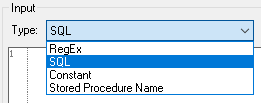


I en brugerdefineret kolonne, kan et inputkriterie angives. Hvis inputkriteriet bliver mødt, vil det angivne output fremgå i den brugerdefinerede kolonne.

For en brugerdefineret kolonne skal angives en input type og en output type.

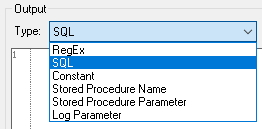
En input type kan være følgende:

* RegEx
* SQL
* Konstant
* Stored Procedure navn



En output type kan være følgende:

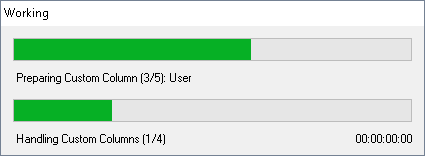
* RegEx
* SQL
* Konstant
* Stored Procedure navn
* Stored Procedure parameter
* Log parameter



Navnet på den brugerdefinerede kolonne skal være unikt, og vil kunne refereres fra andre steder i SQL Event Analyzer. F.eks. er det muligt at tilgå en brugerdefineret kolonne fra en anden brugerdefineret kolonne. Navnet kan ikke være det samme som et statistiknavn f.eks. "AvgDuration" (se afsnittet Statistik for mere information). Ligeledes kan navnet ikke være det samme som navnet på en standard kolonne, f.eks. "Reads".

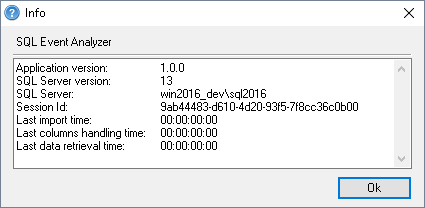
### Generering af output visning

Når hændelser vises i output visningen, er de inden da blevet håndteret udfra de angivne værdier i de brugerdefinerede kolonner. Hvis de angivne værdier i de brugerdefinerede kolonner ændres, vil output visningen blive regenereret.



Regenerering af output visningen kan tage lang tid, alt efter hvor mange brugerdefinerede kolonner der er i kombination med mængden af hændelser.

De forskellige tider for import, kolonnehåndtering og datahentning kan ses i "Hjælp", "Om...":



"Sidste importtid": Tid det har taget at importere de valgte trace filer til databasen.

"Sidste kolonnehåndteringstid": Tid det har taget for de brugerdefinerede kolonner at behandle den importerede data.

"Sidste datahentningstid": Tid det har taget at hente den bearbejdede data fra databasen.

### Input- og output typer

Følgende input typer er tilgængelige for brugerdefinerede kolonner:

**RegEx:** Regular Expression

**SQL:** SQL forespørgsel

**Konstant:** Angiver en konstant værdi

**Stored Procedure navn:** Syntax: 'navn', forekomst

navn er navnet på en Stored Procedure og er værdien angivet efter en "exec" eller en "execute". Hvis "exec" eller "execute" ikke er angivet, vil navnet ikke kunne findes.

forekomst angiver hvilken Stored Procedure, der skal arbejdes med. Normalt vil denne værdi skulle sættes til 1, men hvis der er flere Stored Procedure referencer i samme batch, kan der med denne værdi angives, hvilken en af dem der skal måles på.

Eksempel på output værdi for Stored Procedure navn: 'S\_Proc1', 1

Følgende output typer er tilgængelige for brugerdefinerede kolonner:

**RegEx:** Regular Expression

**SQL:** SQL forespørgsel

**Konstant:** Angiver en konstant værdi

**Stored Procedure navn:** Syntax: forekomst

forekomst angiver hvilken Stored Procedure, der skal arbejdes med. Normalt vil denne værdi skulle sættes til 1, men hvis der er flere Stored Procedure referencer i samme batch, kan der med denne værdi angives, hvilken en af dem der skal returneres.

Eksempel på output værdi for Stored Procedure navn: 1

**Stored Procedure parameter:** Syntax: position, forekomst

position angiver en parameter position til en Stored Procedure.

Eksempel: Hvis der ønskes at finde den 2. parameter (test) fra følgende:

exec S\_Procedure 123, 'test', 456

skal der for position angives 2. Bemærk, at der ved tekstparametre ikke returneres de to apostroffer, der omslutter tekstparameteren.

forekomst angiver hvilken Stored Procedure, der skal arbejdes med. Normalt vil denne værdi skulle sættes til 1, men hvis der er flere Stored Procedure referencer i samme batch, kan der med denne værdi angives, hvilken en af dem der skal måles på.

Eksempel på output værdi for Stored Procedure parameter: 2, 1

**Log parameter:** Angiver en parameter til en log. Værdien skal være en parameter position.

Log information skal have følgende struktur:

-- Log: parameter1, parameter2, ...

Parametre til log information skal opfylde samme krav som parametre til en Stored Procedure.

Eksempel: Hvis der ønskes at finde den 2. parameter (123) fra følgende:

-- Log: 'lhc', 123

skal der i output angives 2.

Hvis "SQL" er valgt som output type, skal SQL forespørgslen opfylde kravene til en SQL sub query. F.eks. er følgende muligt:

select

case TextData

when 'T\_GetCase' then 'View Case'

end

og følgende er ikke muligt:

if TextData = 'T\_GetCase'

begin

select 'View Case'

end

"RegEx", "Stored Procedure navn", "Stored Procedure parameter" og "Log parameter" kan kun benyttes hvis CLR understøttelse er aktiveret på SQL Serveren.

Se mere i afsnittet "CLR funktionalitet sammen med SQL".

Hvis den brugerdefinerede kolonne output type er "SQL", er der en begrænsing på, hvor mange karakterer den returnerede værdi kan være. Hvis den returnerede værdi overstiger 50 karakterer, vil ingen værdier i hele den brugerdefinerede kolonne blive sat (heller ikke værdier, som er mindre end eller lig med 50 karakterer).

Følgende SQL forespørgsel for en "SQL" output type er på præcis 50 karakterer og er valid:

select 12345678901234567890123456789012345678901234567890

Følgende SQL forespørgsel for en "SQL" output type er på 51 karakterer og er invalid:

select 123456789012345678901234567890123456789012345678901

Hvis ovenstående invalide SQL forespørgsel finder sted, vil alle værdier i hele den brugerdefinerede kolonne være tomme.

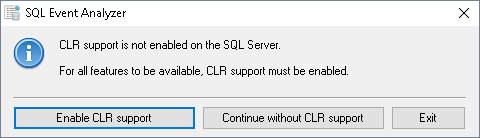
Eksempel på, hvordan den invalide SQL forespørgsel kan omskrives til kun at returnere de første 50 karakterer, og dermed blive valid:

select convert(varchar(50), 123456789012345678901234567890123456789012345678901)

### CLR understøttelse

For at kunne benytte input- og output typer som ikke er "SQL" og "Konstant", skal CLR understøttelse være aktiveret på SQL Serveren.

Som standard er CLR understøttelse ikke aktiveret på en SQL Server. Hvis CLR understøttelse ikke er aktiveret, vil SQL Event Analyzer ved start give mulighed for at aktivere CLR understøttelse:



Det er muligt at vælge at starte SQL Event Analyzer uden CLR understøttelse. Hvis dette vælges, vil følgende input- og output typer ikke være tilgængelige:

* RegEx
* Stored Procedure navn
* Stored Procedure parameter
* Log parameter

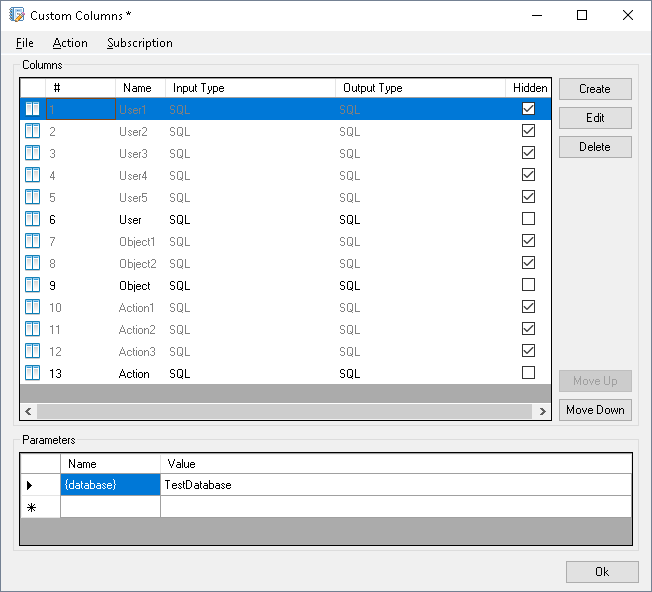
Yderligere er det muligt at aktivere CLR understøttelse midlertidigt ved at holde Shift tasten nede og vælge "Aktiver CLR understøttelse". Hvis CLR understøttelse er midlertidigt aktiveret, vil CLR understøttelse deaktiveres igen når SQL Event Analyzer lukkes eller der skiftes forbindelse til en anden SQL Server.

Bemærk, hvis der er valgt midlertidig CLR understøttelse og SQL Event Analyzer ikke bliver lukket korrekt ved f.eks. strømsvigt, vil CLR understøttelse ikke blive deaktiveret igen.

### Arbejde med brugerdefinerede kolonner

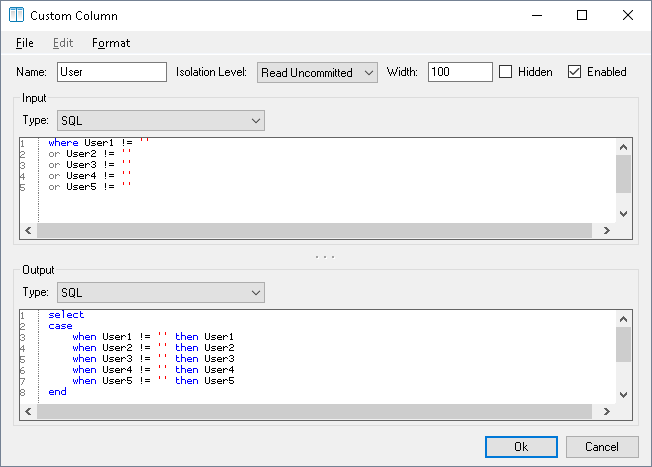
Brugerdefinerede kolonner kan referere hinanden. En reference til en anden brugerdefineret kolonne er kun mulig, hvis den refererede kolonne har et lavere nummer end den nuværende kolonne.

Eksempel:



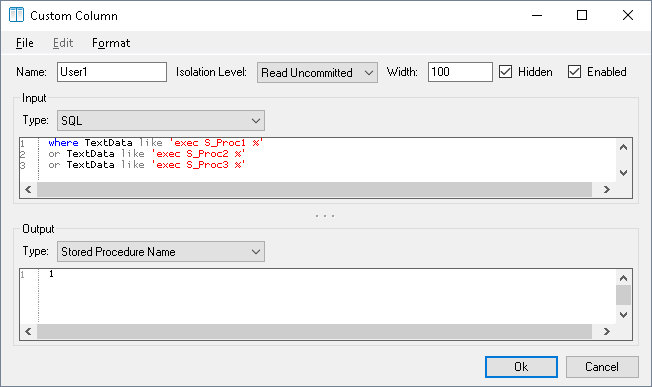
På ovenstående kan kolonne nummer 6 (User) referere kolonnerne fra nummer 1 til 5, men ikke referere kolonne nummer 7 til 13.

I det følgende ses, at kolonne 6 (User) refererer kolonne nummer 1 til 5:

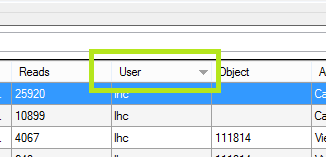


En brugerdefineret kolonne kan altid referere en standard kolonne.

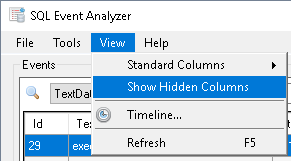
I det følgende ses, at kolonne 1 (User1) refererer standard kolonnen "TextData":



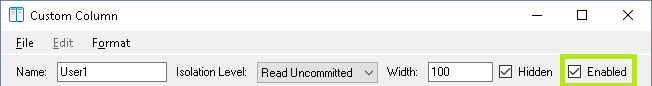
I ovenstående eksempler, er kolonne 6 (User) synlig i output visningen, og kolonne 1 (User1) er skjult:



Det er muligt at vise skjulte kolonner i output visningen ved at vælge "Vis", "Vis skjulte kolonner":

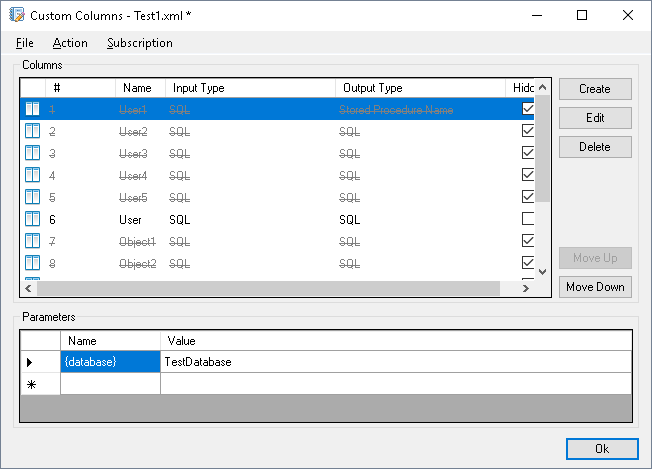


En brugerdefineret kolonne kan markeres som aktiv eller inaktiv:



Hvis en kolonne ikke er markeret som aktiv, er det ikke muligt at referere den fra andre brugerdefinerede kolonner, og den vil ikke blive inkluderet i output visningen.

Inaktive kolonner vises hvor kolonnetekst er gennemstreget:



Hvis CLR understøttelse ikke er aktiveret på SQL Serveren, vil kolonner, der benytter CLR funktionalitet være inaktive, og vil ikke kunne sættes til at være aktive.

### CLR funktionalitet sammen med SQL

Følgende funktioner kan benyttes hvis input- eller output type er sat til "SQL":

* GetRegEx: Returnerer RegEx gruppe 1, der opfylder den angivne RegEx
* MatchRegEx: Returnerer boolsk værdi for om den angivne RegEx kan matches
* GetStoredProcedureName: Returnerer navnet på en funden Stored Procedure i en søgetekst
* MatchStoredProcedureName: Returnerer boolsk værdi for om et angivet Stored Procedure navn findes i en søgetekst
* GetStoredProcedureParameter: Hvis en parameter er fundet på den angivne position, bliver den returneret
* MatchStoredProcedureParameter: Returnerer boolsk værdi for om en Stored Procedure i en søgetekst indeholder en angivet værdi på en angivet position
* GetLogParameter: Hvis en parameter er fundet på den angivne position, bliver den returneret
* MatchLogParameter: Returnerer boolsk værdi for om en log i en søgetekst indeholder en angivet værdi på en angivet position

Input- og output værdier for ovenstående CLR funktioner:

* string GetRegEx(string input, string regExPattern)
* bool MatchRegEx(string input, string regExPattern)
* string GetStoredProcedureName(string input, int occurrence)
* bool MatchStoredProcedureName(string input, string name, int occurrence)
* string GetStoredProcedureParameter(string input, int position, int occurrence)
* bool MatchStoredProcedureParameter(string input, int position, string value, int occurrence)
* string GetLogParameter(string input, int position)
* bool MatchLogParameter(string input, int position, string value)

Følgende eksempler viser forskellig brug af ovenstående CLR funktioner. Der ønskes at returnere alle hændelser fra en TraceData tabel, hvor den første parameter til en Stored Procedure skal være "lhc". Desuden ønskes at den første kolonne i resultatsættet er selve navnet på den fundne Stored Procedure:

select dbo.GetStoredProcedureName(t.TextData, 1), t.\*

from dbo.TraceData\_f63631c1-5824-4dce-b631-f45e187c3cc1 t

where dbo.GetRegEx(t.TextData, 'exec T\_Get.+ ''(.+?)''') = 'lhc'

select dbo.GetStoredProcedureName(t.TextData, 1), t.\*

from TraceData\_f63631c1-5824-4dce-b631-f45e187c3cc1 t

where dbo.MatchStoredProcedureParameter(t.TextData, 1, 'lhc', 1) = 1

select dbo.GetStoredProcedureName(t.TextData, 1), t.\*

from TraceData\_f63631c1-5824-4dce-b631-f45e187c3cc1 t

where dbo.GetStoredProcedureParameter(t.TextData, 1, 1) = 'lhc'

Hvis GetStoredProcedureParameter, MatchStoredProcedureParameter, GetLogParameter eller MatchLogParameter bliver kaldt med 0 som position parameter, vil de returnere følgende:

* GetStoredProcedureParameter: Antal parametre
* MatchStoredProcedureParameter: Boolsk værdi for om antal parametre er lig med value
* GetLogParameter: Antal parametre
* MatchLogParameter: Boolsk værdi for om antal parametre er lig med value

For GetStoredProcedureName, MatchStoredProcedureName, GetStoredProcedureParameter og MatchStoredProcedureParameter gælder, at der skal angives en occurrence parameter. Denne værdi angiver, hvilken Stored Procedure der skal arbejdes med, hvis flere Stored Procedures eksekveres i samme batch. Eksempelvis vil der ved eksekvering af følgende to Stored Procedures i samme batch:

exec s\_proc1

exec s\_proc2

blive returneret s\_proc1 hvis occurrence parameteren er 1, og s\_proc2 hvis occurrence parameteren er 2.

### Renset TextData

Når en eksekveret SQL forespørgsel, TextData, opfanges af SQL Serveren, kan dens formatering være forskellig, selvom det er den samme SQL forespørgsel der eksekveres. Eksempelvis er følgende to SQL forespørgsler ens:

exec s\_proc1

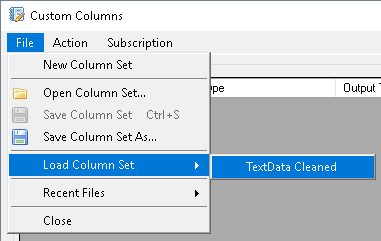
execute [s\_proc1]

SQL Event Analyzer indeholder funktionalitet til at "rense" forespørgslerne, således at de fremstår så ens som muligt. Ovenstående to SQL forespørgsler vil efter rensning fremstå som:

s\_proc1

Dette muliggør f.eks. statistik på, hvor mange gange en SQL forespørgsel er eksekveret.

Renset TextData funktionaliteten kan benyttes ved at vælge "Hent kolonnesæt", "Renset TextData" under "Brugerdefinerede kolonner":



### Eksempel på oprettelse af index på en brugerdefineret kolonne

Det kan være ønsket, at kunne operere på enten DML eller DDL fra de brugerdefinerede kolonner. Dette kan lade sig gøre ved at bryde funktionaliteten i SQL output typen.

Brydningen af SQL output typen sker ved at indsætte følgende:

'')

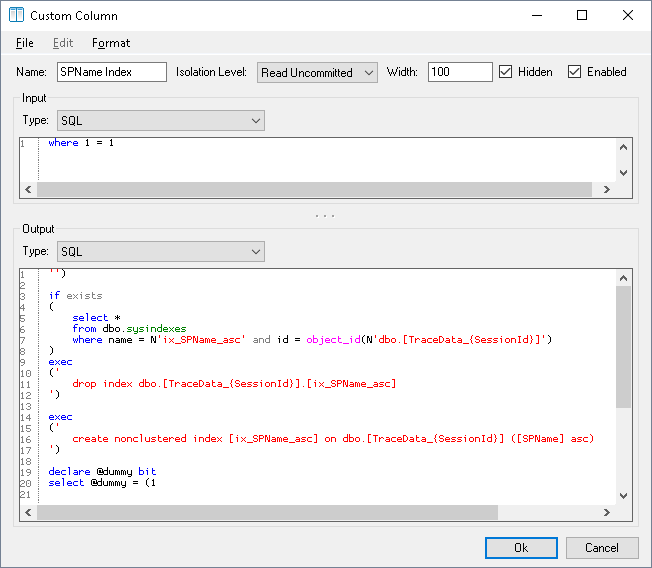
...

declare @dummy bit

select @dummy = (1

Hvor ... indeholder det SQL udtryk der ønskes eksekveret.

I det følgende vises et eksempel på oprettelse af et index på en brugerdefineret kolonne med navnet SPName:



Bemærk, at i ovenstående benyttes parameteren {SessionId}, som erstattes af den korrekte session id under den endelige kørsel.

SQL udtrykket i ovenstående skærmbillede er følgende:

'')

if exists

(

select \*

from dbo.sysindexes

where name = N'ix\_SPName\_asc' and id = object\_id(N'dbo.[TraceData\_{SessionId}]')

)

exec

('

drop index dbo.[TraceData\_{SessionId}].[ix\_SPName\_asc]

')

exec

('

create nonclustered index [ix\_SPName\_asc] on dbo.[TraceData\_{SessionId}] ([SPName] asc)

')

declare @dummy bit

select @dummy = (1

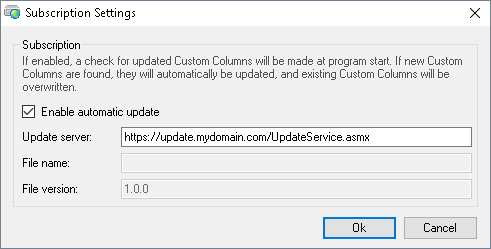
### Abonnement

Det er muligt at abonnere på brugerdefinerede kolonner, og automatisk nedhente nye versioner af et brugerdefineret kolonnesæt fra en central opdateringsserver.

Abonnementfunktionaliteten aktiveres fra "Abonnement", "Indstillinger...":



Her kan angives sti til opdateringsserver samt om funktionaliteten skal aktiveres:



Abonnementfunktionaliteten er tilknyttet det aktive brugerdefinerede kolonnesæt, d.v.s. det f.eks. er muligt at angive en anden opdateringsserver for et andet brugerdefineret kolonnesæt.

Signatur for web service kommunikation:

public string GetDownloadUrl(string productName, string machineName, string userName, string domainName, string allowUpdateCriteria)

public string GetLatestVersion(string productName, string currentVersion, string machineName, string userName, string domainName, string allowUpdateCriteria)

public string GetChangelogUrl(string productName, string allowUpdateCriteria)

Changelog format er:

1.0.1

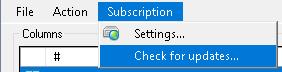
Feature X implemented.

1.0.0

Initial version.

Når abonnementfunktionaliteten er aktiveret, vil der ved programstart, samt ved indlæsning af ny fil med brugerdefinerede kolonner, blive tjekket for, om der findes en nyere version af de pågældende brugerdefinerede kolonner.

Det er også muligt manuelt at tjekke for opdateringer fra "Abonnement", "Check for opdateringer...":



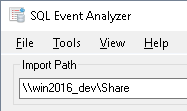
Hvis der er opdateringer til de brugerdefinerede kolonner, vil det fremgå af menulinien:



De nye brugerdefinerede kolonner vil først træde i kraft ved genstart af applikationen.

## Hændelser fra tracefiler

Tracefiler hentes fra den angivne sti under "Import sti":



For at en tracefil kan importeres i SQL Event Analyzer, skal tracefilen indeholde minimum følgende data:

* TextData
* EventClass
* StartTime

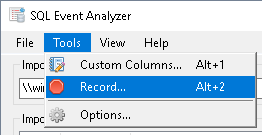
Eksempelvis kan en tracefil fra SQL Server Profiler importeres.

Hvis SQL Server 2012 eller nyere benyttes, kan Extended Event Trace filer (.xel) også importeres.

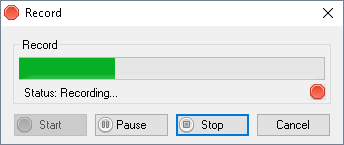
## Realtime hændelser

I stedet for importering af hændelser fra en tracefil, kan hændelser optages i realtime.

En optagelse startes fra "Værktøjer", "Optag...":



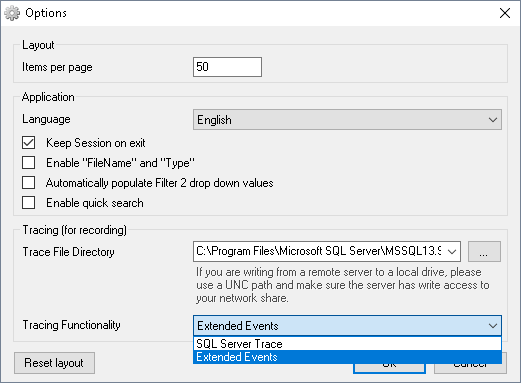
Når en optagelse er i gang, registreres alle hændelser der modtages på SQL Serveren.



### Ændring af tracing funktionalitet

Hvis SQL Server 2012 eller nyere anvendes, er det muligt at ændre den tracing funktionalitet, der benyttes til optagelse af realtime hændelser.

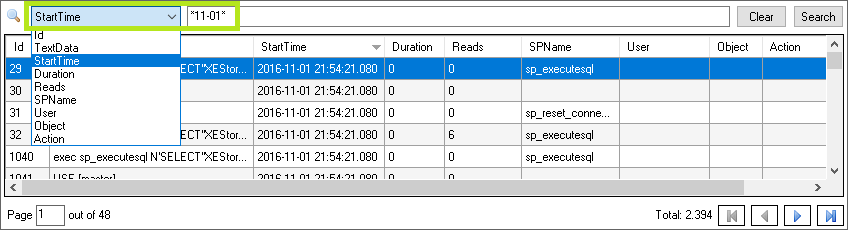
Hvis SQL Event Analyzer startes på en SQL Server 2012 eller nyere, vil tracing funktionalitet som standard være sat til "Extended Events". Dette kan ændres i "Indstillinger" menuen:



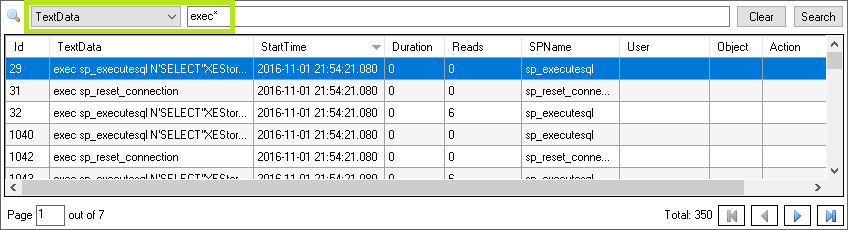
Bemærk, at det ikke er muligt at vælge "Extended Events", hvis der anvendes en SQL Server der er ældre end SQL Server 2012.

## Søgning og sortering

Det er muligt at søge i alle kolonner. Søgning understøtter wildcard søgning ved brug af \*. Med undtagelse af "TextData" kolonnen, kan wildcards benyttes både før- og efter søgeordet:

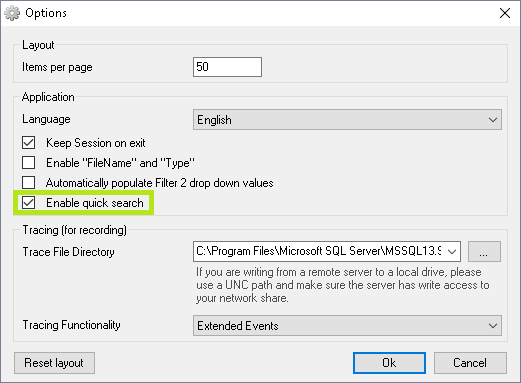


Ved søgning i "TextData" feltet, kan wildcards kun benyttes efter søgeordet:



Alle kolonner kan sorteres, så man f.eks. kan se hvilken hændelse der har haft den største eksekveringstid.

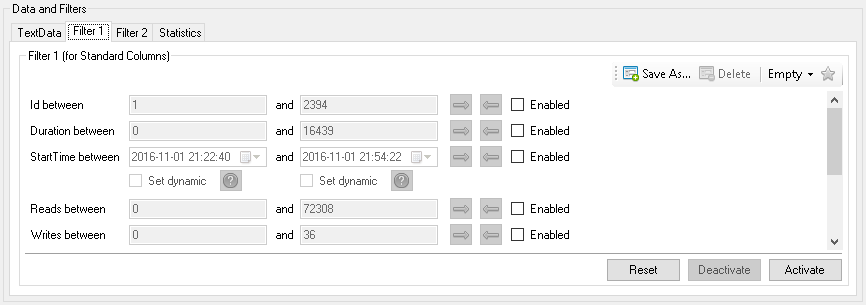
Bemærk, den viste søgning kun er muligt, hvis der i indstillinger er aktiveret "Aktiver hurtigsøgning":



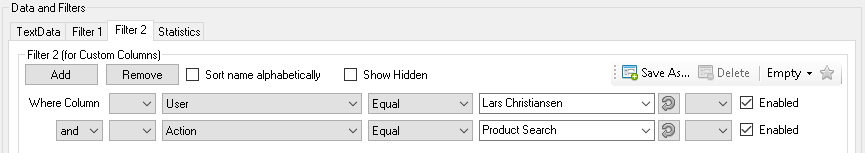
## Filtre

Det er muligt at filtrere hændelser ved hjælp af to filtre:

Filter 1: Filtrering af data fra standard kolonner:



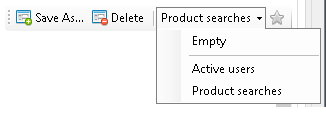
Filter 2: Filtrering af data fra brugerdefinerede kolonner:



Når et filter er aktivt, vil filternavnet være markeret med en \*:



Søgninger defineret i Filter 1 og Filter 2 kan gemmes, så de kan genbruges. For at gemme en søgning, vælg "Gem som..." og angiv et navn. Herefter vil den gemte søgning kunne findes i listen som vist på følgende:



Ligeledes kan søgninger slettes og redigeres.

Gemte søgninger kan markeres som systemobjekter. Hvis en gemt søgning er markeret som systemobjekt, vil søgningen ikke kunne slettes, redigeres eller overskrives.

Gemte søgninger kan markeres som systemobjekter ved at tilføje navnene på de gemte søgninger i registreringsdatabasenøglen for SQL Event Analyzer. Navne på gemte søgninger skal adskilles med | på følgende måde:



Navnet på den gemte søgning kan ikke indeholde følgende tegn:

,'&[]

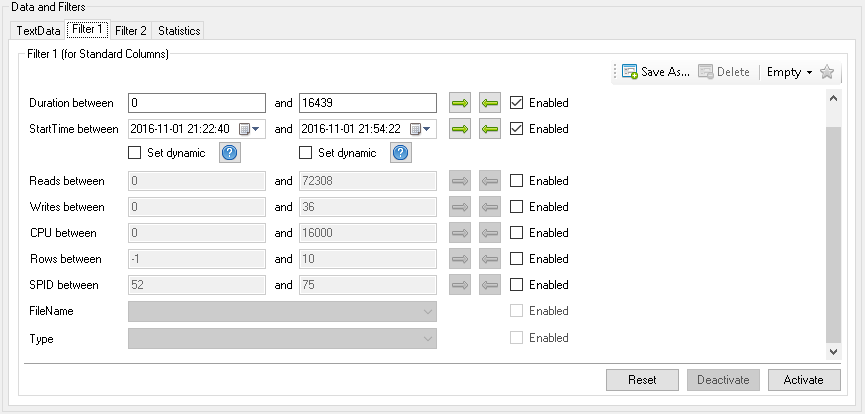
Ligeledes kan navnet ikke være "Empty" eller indeholde tegn, som er ugyldige i filnavne.

### Filter 1

Værdierne i filteret vil ved start være udfyldt med minimum, maksimum og unikke værdier for data. Eksempelvis vil den første "StartTime" værdi være udfyldt med tidspunktet for den første af hændelserne, og den anden "StartTime" værdi vil være udfyldt med tidspunkt for den sidste af hændelserne.

Ligeledes vil f.eks. "Type" være udfyldt med unikke værdier for "Type" der findes blandt hændelserne.

Som standard, er "FileName" og "Type" deaktiveret:

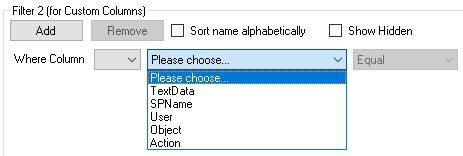


Disse kan aktiveres i "Indstillinger" menuen. Bemærk, at hvis "FileName" og "Type" aktiveres, vil det tage længere tid at arbejde med TraceData tabellerne. Hvis "FileName og "Type" ikke er aktiveret, vil værdierne stadig blive gemt i TraceData tabellerne.

Ved tryk på "Nulstil" knappen, vil værdierne blive sat til de samme værdier, som ved start.

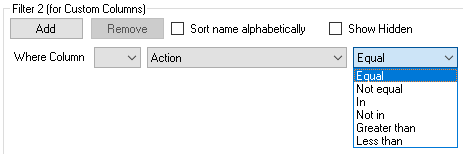
### Filter 2

Det er muligt at tilføje filtre på brugerdefinerede kolonner. Først vælges den brugerdefinerede kolonne:

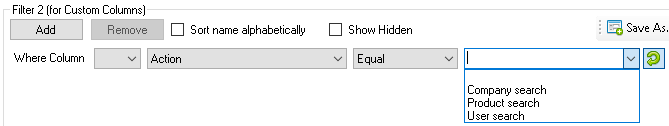


Bemærk, "TextData" vil altid stå øverst og er ikke påvirket af "Sorter navn alfabetisk".

Kriterie vælges:

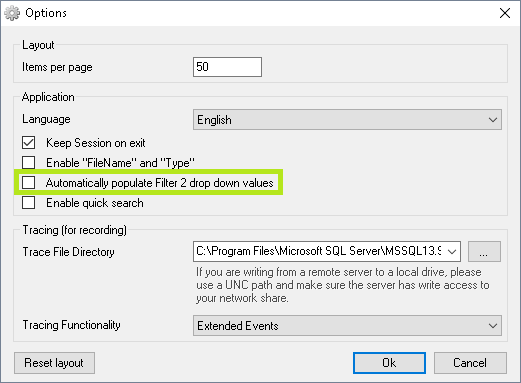


Værdi vælges eller angives:

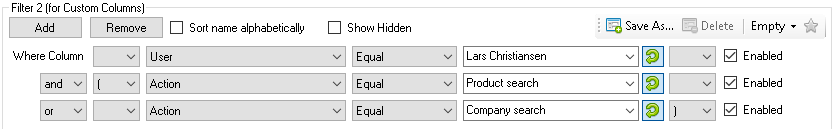


I listen over værdier er det muligt selv at angive en værdi, eller vælge en af de unikke værdier tilhørende den brugerdefinerede kolonne. De unikke værdier vises altid i alfabetisk rækkefølge.

Bemærk, som standard er de unikke listeværdier ikke sat til at blive automatisk udfyldt. Denne værdi kan ændres i "Indstillinger" menuen:



Det er muligt at tilføje så mange søgekriterier man vil:



Bemærk brugen af paranteser i ovenstående.

Ved brug af "Lig med" og "Ikke lige med" operatorerne, kan benyttes wildcards. Der kan benyttes wildcards både før- og efter søgeværdien:



Som wildcard kan benyttes følgende karakterer: "%" og "\*".

Ved brug af "Findes i" og "Ikke findes i" operatorerne, kan der søges efter en række af værdier. Værdierne skal være adskilt af komma:

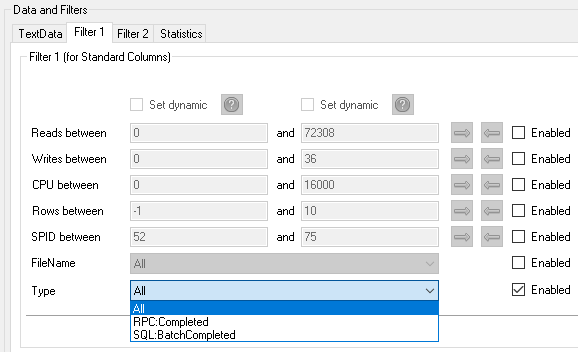


Ved brug af "Aktiv", kan angives om den pågældende filtrering medtages i filteret eller ej.

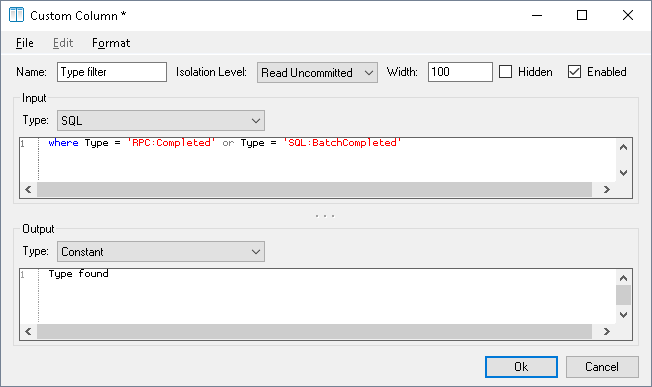
### Eksempel på filtrering

For en række hændelser ønskes fundet alle hændelser, der har en "Type" som enten er "RPC:Completed" eller "SQL:BatchCompleted".

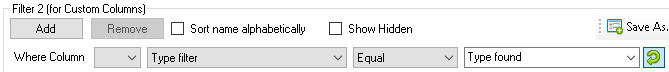
I Filter 1 er det muligt at filtrere på "Type", men her kan kun vælges en enkelt værdi ad gangen:



Filter 2 kan ikke filtrere på standard kolonner, så opgaven må løses ved oprettelse af en brugerdefineret kolonne:



I Filter 2 kan nu vælges værdien fra den brugerdefinerede kolonne og filtreringen kan gennemføres:



## Kørsel i uovervåget tilstand

SQL Event Analyzer kan køres uovervåget. Uovervåget tilstand aktiveres ved at starte SQL Event Analyzer med følgende kommandolinieparametre:

SQLEventAnalyzer.exe [

[

(-p:"Post script fil")

(-e:"Post script")

(-z:"Antal filer")

(-d)

(-g:"Statistiknavne" (-f1:"Filter 1 navne") (-f2:"Filter 2 navne") [(-sp:"Sti") eller (-sw:"Statistiknavne")])

[

[

[

[-n:Antal tracefiler] eller [-f:"Tracefilnavn"]

]

(-s:"Session Id (-x))

]

eller

[-i:"Session Id" (-x)]

]

[-l:"Import sti"]

]

eller

[-u:"Session Id" (-v (-a:"Applikationsnavn")) (-x)]

eller

[-r]

]

(-c:"Brugerdefineret kolonnesæt filnavn")

(-t:"SQL Server Connection String" (-ms:"SQL"))

(-o)

(-w:"Web service")

(-b:"Databasenavn")

(-w:"Web service")

(-b:"Databasenavn")

(-m:"Tekst")

[]: Obligatorisk parameter

(): Valgfri parameter

Parametre:

(-p:"Post script fil"):

Fuld sti til SQL script fil.

Scriptet vil blive eksekveret efter import af tracefiler.

(-e:"Post script"):

SQL script.

Scriptet vil blive eksekveret efter import af tracefiler.

(-z:"Antal filer"):

Komprimer tracefiler efter import. De seneste "Antal filer" vil blive beholdt og resten vil blive slettet.

Brug 0 for at beholde alle filer.

(-d):

Slet tracefiler efter import.

[-n:Antal tracefiler]:

Importerer de N nyeste tracefiler.

Brug 0 for at importere alle tracefiler.

[-f:"Tracefilnavn"]:

Importerer alle tracefiler nyere end den angivne tracefil.

Hændelserne i den angivne tracefil vil ikke blive importeret.

(-s:"Session Id"):

Tilføj hændelser til den angivne session.

Hvis -s er angivet, vil hændelser blive tilføjet til den angivne session.

Hvis -s ikke er angivet, vil hændelser blive importeret til en ny session.

[-i:"Session Id"]:

Importerer alle tracefiler nyere end den sidst importerede tracefil fundet i "FileName" kolonnen i TraceData tabellen for den angivne session.

Hvis -i er angivet, vil hændelser blive tilføjet til den angivne session.

[-l:"Import sti"]:

Sti til tracefiler, som skal importeres.

[-u:"Session Id"]:

Benyt den angivne session ved start.

(-v):

Deaktiver mulighed for at ændre data i TraceData tabellen.

(-a:"Applikationsnavn"):

Benyt angivet applikationsnavn i stedet for "SQL Event Analyzer".

(-b:"Databasenavn"):

Benyt angivet databasenavn i stedet for "SQLEventAnalyzer".

(-ms:"SQL"):

Tilføj output fra angivet SQL til log for kørsel. Output returneres fra første række i kolonnen "Message".

-t parameteren angiver den benyttede SQL Server forbindelse.

(-m:"Tekst"):

Tilføj angivet tekst til log for kørsel.

(-ms:"SQL"):

Tilføj output fra angivet SQL til log for kørsel. Output returneres fra første række i kolonnen "Message".

-t parameteren angiver den benyttede SQL Server forbindelse.

(-x):

Gennemtving sletning af TraceData tabellen for den angivne session ved afslutning.

[-r]:

Start i optagetilstand.

(-c:"Brugerdefineret kolonnesæt filnavn"):

Fuld sti til fil, som skal benyttes som brugerdefineret kolonnesæt.

(-t:"SQL Server Connection String"):

Forbind ved brug af den angivne SQL Server Connection String.

Hvis et password benyttes i den angivne SQL Server Connection String, skal det være krypteret.

Hvis -t ikke er angivet, vil den sidst succesfulde forbindelse blive benyttet.

(-o):

Log for kørsel gemmes i eksekveringssti.

(-w:"Web service"):

Send log for kørsel til den angivne web service.

(-g:"Statistiknavne"):

Generer statistik udfra de gemte statistikker med de angivne navne.

Navne kan adskilles med komma.

(-f1:"Filter 1 navne"):

Aktiver Filter 1 med de angivne navne til brug for generering af statistik.

Navne kan adskilles med komma.

(-f2:"Filter 2 navne"):

Aktiver Filter 2 med de angivne navne til brug for generering af statistik.

Navne kan adskilles med komma.

(-sw:"Statistiknavne"):

Kræver at -w er angivet. Sender genereret statistik med de angivne navne til den angivne web service.

De angivne navne skal skal være repræsenteret i navnene fra -g parameteren.

(-sp:"Sti")

Gemmer genereret statistik i den angivne sti.

Eksempel 1 (importer de to nyeste tracefiler og kør script bagefter):

SQLEventAnalyzer.exe -p:"C:\SomeDir\PostScript.sql" -n:2 -l:"C:\TraceFiles"

Eksempel 2 (importer alle tracefiler nyere end "LastImported.trc"):

SQLEventAnalyzer.exe -f:"C:\SomeDir\LastImported.trc" -l:"C:\TraceFiles"

Eksempel 3 (importer alle tracefiler nyere end den sidst importerede tracefil fundet i "FileName" kolonnen i TraceData tabellen og tilføj hændelserne til en eksisterende session):

SQLEventAnalyzer.exe -i:"collect" -l:"C:\TraceFiles"

Post script filen og post scriptet er valgfrie, og kan enten være tomme eller indeholde en SQL forespørgsel, som vil blive eksekveret efter TraceData tabellen er blevet oprettet, og brugerdefinerede kolonner er blevet genereret. SQL forespørgslen eksekveres i samme database, hvor TraceData tabellen findes.

Post script filen og post scriptet kan benytte parametre efter samme princip som beskrevet i afsnittet "Parametre". Eksempelvis kan {SessionId} eller valgfrie parametre benyttes.

Det er muligt både at angive -p og -e parametrene samtidigt. Hvis dette er tilfældet, vil post script filen angivet med -p blive eksekveret først, og post script'et angivet med -e blive eksekveret derefter.

"Tracefilnavn" kan være filer af typen SQL Server Trace (.trc) eller, hvis understøttet, Extended Event Trace filer (.xel).

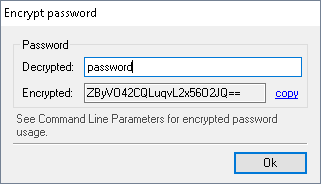
Parameteren -u kan bruges til at benytte en angivet session direkte ved start af SQL Event Analyzer. Dette kan være anvendeligt i situationer, hvor der kun ønskes adgang til handlinger og hvor import af tracefiler ikke er relevant. Ydermere, kan parameteren -v angives sammen med -u for at starte SQL Event Analyzer i en tilstand, hvor det kun er muligt at se handlinger, men hvor funktionalitet der kan ændre handlinger er deaktiveret.

Følgende funktionalitet er deaktiveret ved brug -v parameteren:

* Menuen "Brugerdefinerede kolonner"
* Menuen "Skift forbindelse"
* Optagefunktionalitet
* "Tilbage" knappen
* "Vis skjulte kolonner"

Hvis parameteren -z er angivet, vil den importerede tracefil blive komprimeret til et 7zip arkiv.

Hvis parameteren -t er angivet, kan der i den angivne SQL Server Connection String vælges at angive et password. Dette password skal være krypteret. Et password kan krypteres ved at vælge "Hjælp", "Kommandolinieparametre...", "Værktøjer", "Generer krypteret password...":



Hvis parameteren -m er angivet, vil den angivne brugerdefinerede tekst blive tilføjet sidst i log for kørslen.

Output vil blive tilføjet til log for kørslen i følgende format:

2015-05-18T08:48:43: Custom Message:

[BEGIN Custom Message]

...

[END Custom Message]

Hvis parameteren -ms er angivet, vil output fra den angivne SQL blive tilføjet sidst i log for kørslen. Output-kolonnenavn skal være "Message" (uden anførselstegn). Output returneres kun fra den første række.

Output vil blive tilføjet til log for kørslen i følgende format:

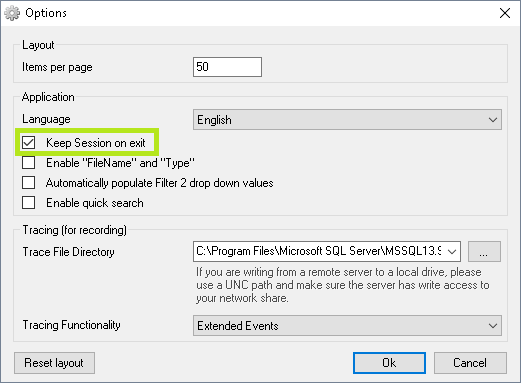
2015-05-18T08:48:43: Custom SQL Message:

[BEGIN Custom SQL Message]

...

[END Custom SQL Message]

Bemærk, den angivne session vil blive slettet, hvis ikke "Behold session ved afslutning" er aktiveret i "Indstillinger":



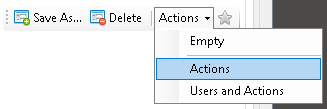
Hvis parametrene -u, -i eller -s er angivet, vil sessionen ikke blive slettet selvom "Behold session ved afslutning" ikke er aktiveret. Der kan tvinges en sletning af sessionen igennem, ved at angive parameteren -x.

### Autogenerering af aggregeret statistik

Hvis parameteren -g er angivet sammen med -sp og/eller -sw, vil der blive genereret aggregeret statistik (se mere i afsnittet Statistik).

Hvis parameteren -sw er angivet sammen med -g parameteren, vil statistikken blive sendt til den web service, der er angivet ved -w parameteren (For yderligere information se afsnittet Send kørselslog til web service). Hvis -sp er angivet, vil statistikken blive gemt i den angivne sti med filnavnet Statistics.zip.

Den statistik, der skal benyttes til autogenereringen, bestemmes udfra det angivne navn fra -g parameteren. Navnet skal stemme overens med en gemt søgning fra statistik fanebladet:



Her kan sammensættes den ønskede gruppering af kolonner til brug for aggregeringen. For at benytte ovenstående sammensætning, "Actions", skal -g parameteren angives på følgende måde:

-g:"Actions"

Hvis der ønsked at generere flere statistikker for flere gemte søgninger, kan disse angives ved brug af komma:

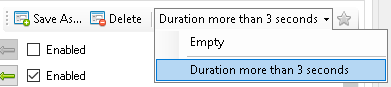
-g:"Actions, Users and Actions"

Det er muligt at benytte Filter 1 og Filter 2 til at filtrere datagrundlaget til brug for generering af statistikken (se mere i afsnittet Filtreret statistik).

Filtre angives ved brug af -f1 parameteren for Filter 1 og -f2 parameteren for Filter 2. Eksempel på brug af Filter 1:

-f1:"Duration more than 3 seconds"

Navnet på det angivne Filter 1, skal ligeledes findes som en gemt søgning for Filter 1:



Hvis flere statistikker er angivet i -g parameteren, kan der ligeledes i -f1 parameteren og -f2 parameteren angives hvilke filtre, der evt. skal benyttes til den pågældende statistik. Filternavne adskilles af komma. Eksempel:

-f2:"Customer Actions, Users in Department ABC"

Eksempel på benyttelse af tre statistikker, hvor der skal være Filter 1 på den første statistik og den sidste statistik:

-g:"SomeStat1, SomeStat2, SomeStat3" -f1:"SomeFilter1,,SomeFilter2"

Den genererede statistik bliver gemt i CSV filformat, der kan åbnes direkte i Microsoft Excel. Filerne navngives i følgende format:

[Statistiknavn][Periode start][Periode slut][Filter 1 navn][Filter 2 navn][TotalRows].CSV

[Periode start] og [Periode slut] angiver start- og sluttidspunkt for handlinger der udgør perioden, der ligger til grundlag for statistikken.

[TotalRows] angiver den totale mængde handlinger, der ligger til grundlag for statistikken.

Den endelige statistik bliver leveret i komprimeret format i en fil med navnet Statistics.zip. Statistics.zip indeholder de ønskede statistikker i CSV filformat.

Den eksporterede statistik vil indeholde alle kolonner, lige meget hvilke kolonner der er sat til at blive vist i statistik visningen. Kolonnerækkefølgen bestemmes ud fra rækkefølgen af de gemte statistiksøgninger, og ikke ud fra den viste kolonnerækkefølge som ved eksportering af statistik fra brugergrænsefladen.

Eksempel på autogenerering af to statistikker med det samme navn ("Actions"), hvor to forskellige Filter 1 ("All days" og "Last week") er pålagt, hvor statistikken med "Last week" skal sendes til web service og begge statistikker skal gemmes til en sti:

-g:"Actions, Actions" -f1:"All days, Last week" -sw:", Actions" -sp:"C:\somedir"

Bemærk parameteren:

-sw:", Actions"

hvor det angives, at det er nummer to genereret statistik (som passer til Filter 1 "Last week"), der skal sendes til web servicen.

### Kørsel i service context

Hvis SQL Event Analyzer køres i uovervåget tilstand via f.eks. Windows Task Scheduler, vil eksekveringen foregå i Task Scheduleren's service context.

Ved eksekvering i en service context, vil den grafiske brugergrænseflade ikke blive vist. Logs fra kørsel i en service context vil kunne findes i stien hvor SQL Event Analyzer er installeret.

Logfiler fra eksekvering i en service context gemmes fra de seneste 10 eksekveringer. Ved den 11. eksekvering vil den ældste af de 10 logfiler blive slettet. Eksempel på logfilnavn:

SQL Event Analyzer 25082014-110831.log

SQL Event Analyzer detekterer automatisk, om den eksekveres i en service context eller ej. Hvis SQL Event Analyzer eksempelvis startes fra en kommandoprompt via en uovervåget kørsel, vil den grafiske brugergrænseflade blive vist.

Ved kørsel i en service context, vil parameteren -o blive ignoreret, da der altid genereres kørselslog i en service context kørsel.

### Returkoder fra uovervåget kørsel

Ved eksekvering i uovervåget tilstand, returnerer SQL Event Analyzer en kode ved afsluttet eksekvering.

Betydning af returkoderne:

|  |  |
| --- | --- |
| Returkode | Betydning |
| 0 | Eksekvering succesfuld |
| -1 | Fejl ved kørsel i uovervåget tilstand, som ikke er startet med kommandolinieparameter -r eller -u |
| -2 | Fejl ved kørsel i optagetilstand (Kommandolinieparameter: -r) |
| -3 | Fejl ved kørsel med benyttelse af angivet session ved start (Kommandolinieparameter: -u) |
| -4 | Fejl i post script eksekvering |
| -5 | Fejl ved sletning af importerede tracefiler |
| -6 | Fejl ved import af tracefiler |
| -7 | Fejl ved komprimering af importerede tracefiler |
| -8 | Fejl ved kommunikation med web service |
| -9 | Fejl ved generering af statistik |
| -10 | Fejl ved eksekvering af SQL script til brug for output til log for kørsel (Kommandolinieparameter: -ms) |

### Send kørselslog til web service

Log for kørsel kan ved angivelse af -w parameteren sendes til en web service.

Signatur for web service kommunikation uden vedhæftet fil:

public bool RegisterLogEvent(string productName, string currentVersion, string machineName, string userName, string domainName, int statusCode, string message)

Signatur for web service kommunikation med vedhæftet fil (benyttes hvis -g og -sw parametrene er brugt. Se mere i afsnittet Autogenerering af aggregeret statistik):

public bool RegisterLogEventWithAttachment(string productName, string currentVersion, string machineName, string userName, string domainName, int statusCode, string message, byte[] attachment)

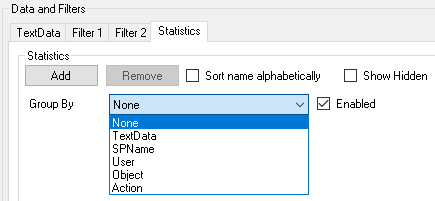
## Kørsel i optagetilstand

SQL Event Analyzer kan sættes til at starte direkte op i optagetilstand. Optagetilstand aktiveres ved at starte SQL Event Analyzer med kommandolinieparametren -r :

SQLEventAnalyzer.exe -r

## Statistik

Statistik om de indlæste hændelser kan tilgåes fra "Statistik" fanebladet:



Her er det muligt at få vist statistisk data udfra en valgfri gruppering af data.

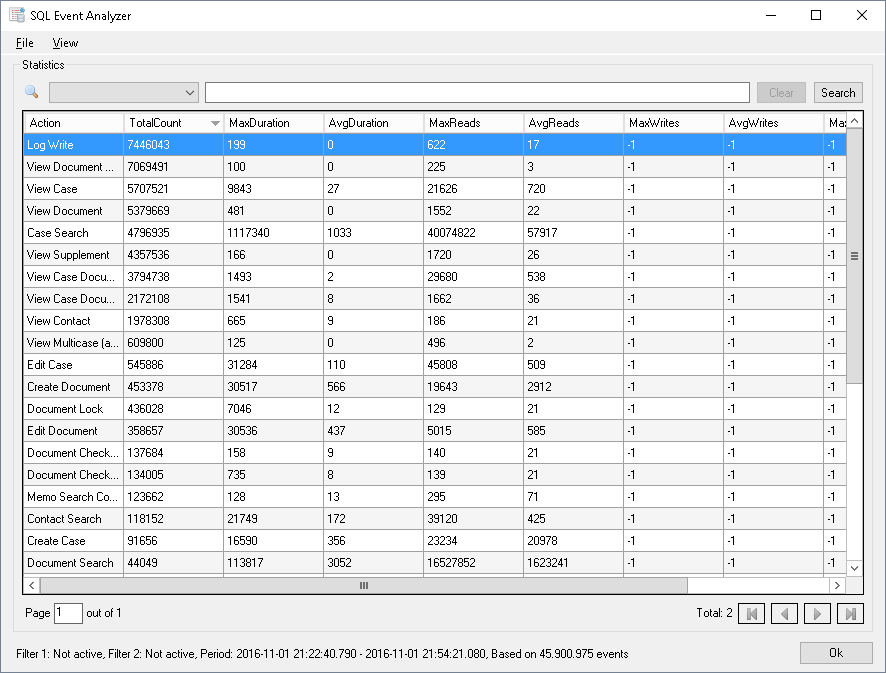
Hvis ingen gruppering vælges, vises den samlede statistik for alle hændelser.

Bemærk, "TextData" vil altid stå øverst og er ikke påvirket af "Sorter navn alfabetisk".

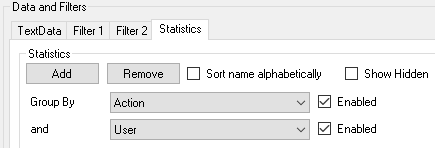
Som ved Filter 1 og Filter 2, kan valgte søgninger gemmes, slettes og redigeres (for yderligere information, se afsnittet Filtre).

Eksempel:

Visning af statistik grupperet efter den brugerdefinerede kolonne "Action":



Det er muligt at gruppere på så mange brugerdefinerede kolonner som ønskes:

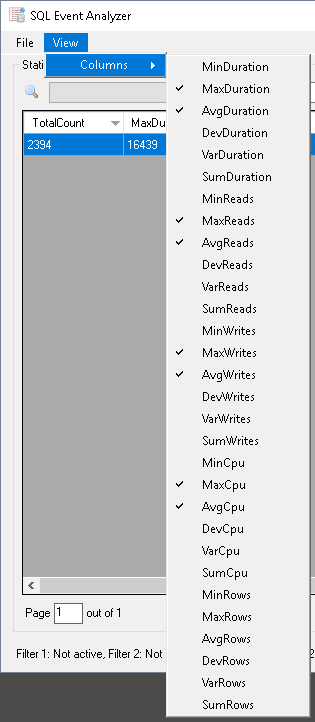


Følgende statistiske data kan vises:

* TotalCount: Antal
* MinDuration: Minimum eksekveringstid
* MaxDuration: Maksimum eksekveringstid
* AvgDuration: Gennemsnitlig eksekveringstid
* DevDuration: Standardafvigelse i eksekveringstid
* VarDuration: Varians i eksekveringstid
* SumDuration: Sum af eksekveringstid
* MinReads: Minimum antal Reads
* MaxReads: Maksimum antal Reads
* AvgReads: Gennemsnitligt antal Reads
* DevReads: Standardafvigelse i antal Reads
* VarReads: Varians i antal Reads
* SumReads: Sum af antal Reads
* MinWrites: Minimum antal Writes
* MaxWrites: Maksimum antal Writes
* AvgWrites: Gennemsnitligt antal Writes
* DevWrites: Standardafvigelse i antal Writes
* VarWrites: Varians i antal Writes
* SumWrites: Sum af antal Writes
* MinCPU: Minimum antal CPU
* MaxCPU: Maksimum antal CPU
* AvgCPU: Gennemsnitligt antal CPU
* DevCPU: Standardafvigelse i antal CPU
* VarCPU: Varians i antal CPU
* SumCPU: Sum af antal CPU
* MinRows: Minimum antal Rows
* MaxRows: Maksimum antal Rows
* AvgRows: Gennemsnitligt antal Rows
* DevRows: Standardafvigelse i antal Rows
* VarRows: Varians i antal Rows
* SumRows: Sum af antal Rows

Eksekveringstider er angivet i millisekunder.

Ikke alle ovenstående kolonner vises som standard. Det kan vælges, hvilke kolonner der skal vises, fra "Vis", "Kolonner":



### Filtreret statistik

Det er muligt at benytte både Filter 1 og Filter 2 til at filtrere det statistiske datagrundlag.

Nederst i venstre hjørne på statistikvisningssiden, ses hvilke filtre der er aktiveret:



### Performanceforbedring ved arbejde med statistik

Hvis der arbejdes med store datamængder, vil der ved fordel kunne oprettes et "Column Store Index" på TraceData tabellen. Dette kræver at der benyttes SQL Server 2012 eller nyere.

Et "Column Store Index" kan oprettes manuelt ved at eksekvere følgende:

create nonclustered columnstore index ix\_CS on dbo.[TraceData\_{SessionId}] (ID, [User], [Object], [Action], Duration, Reads, Writes, CPU, Rows)

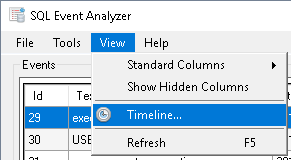
Hvor {SessionId} skal erstattes af det ønskede session id.

I ovenstående script er angivet de brugerdefinerede kolonner [User], [Object], [Action]. Disse kan frit ændres til hvad der ønskes.

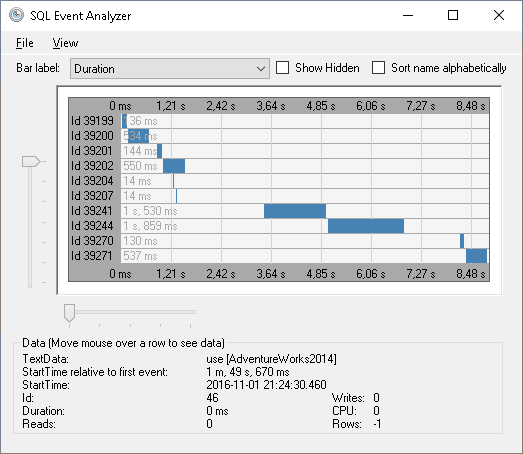
Navngivningen af indexet skal være ix\_CS. Hvis et andet navn benyttes, vil SQL Event Analyzer fejle.

## Tidslinie

En tidslinie over hændelser kan tilgåes fra "Vis", "Tidslinie...":

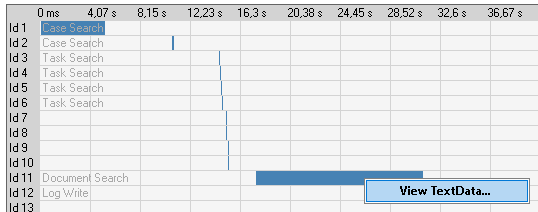


Tidslinien:



Information om hændelser kan ses ved at flytte musen henover de enkelte hændelser.

Det er muligt at højreklikke på en hændelse og få vist hændelsens SQL udtryk:

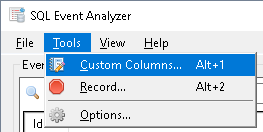


Bemærk, at en tidslinie ikke kan vises for mere end 1000 hændelser, og maksimalt kan vise hændelser der har et tidsinterval på 1 dag.

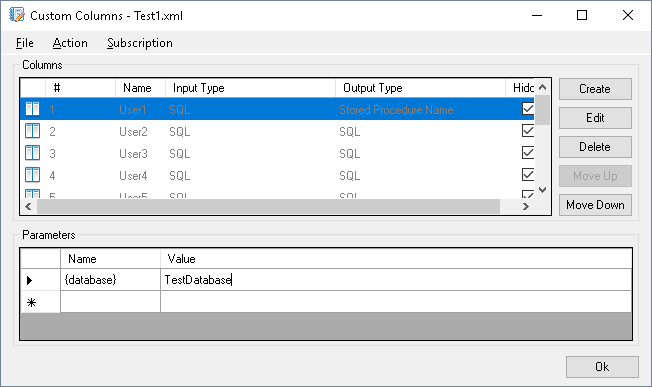
## Parametre

Parametre kan benyttes til at erstatte værdier i teksten for input- og output typer.

Parametre tilgåes fra "Værktøjer", "Brugerdefinerede kolonner...":

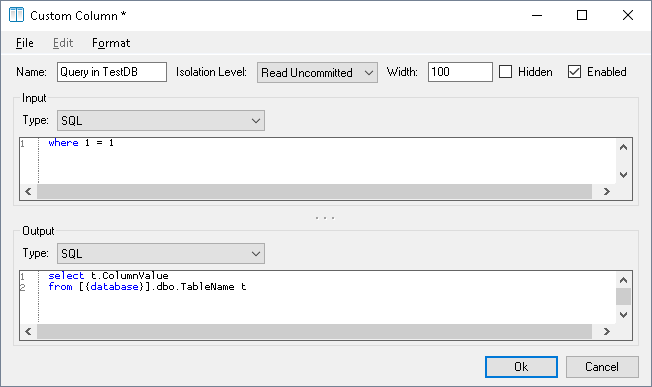


I "Parametre" sektionen, kan parametre tilføjes, redigeres og slettes:



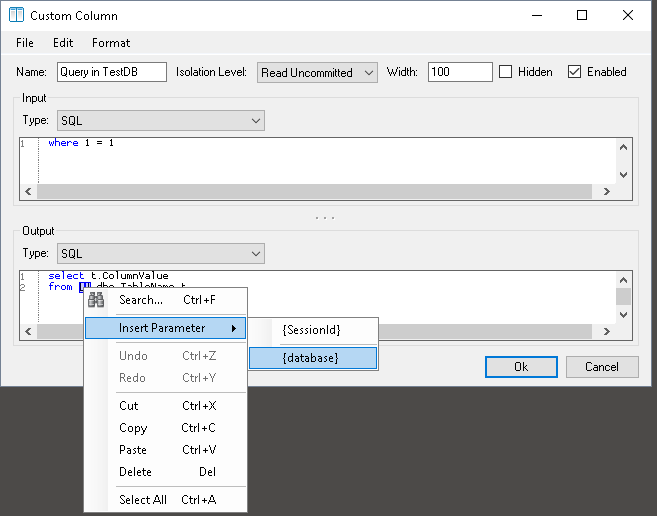
Værdien angivet i "Navn" vil under kørsel blive erstattet med værdien angivet i "Værdi". I ovenstående eksempel vil "{database}" blive erstattet med "TestDatabase".

Følgende viser et eksempel på brug af en parameterværdi i output teksten:



Bemærk, at værdier for parameter "Navn" og "Værdi" kan angives valgfrit, og behøver ikke at være angivet med { }.

Angivede parametre kan indsættes direkte fra højrekliksmenuen:



{SessionId} parameteren er en standard parameter, som indeholder den aktive session id.