

# Probleme im Griff mit Extended Events

Autor Till Lohfink

Datum der Veröffentlichung 29.09.2017

Die Verwendung ist nur für den persönlichen Gebrauch und nur im Rahmen der Nutzung der Bissantz-Softwareprodukte gestattet. Für die Richtigkeit des Inhalts wird keine Haftung übernommen. Jedwede Weitergabe, intern oder an Dritte, und die Veröffentlichung sind ausdrücklich untersagt. Sämtliche Unterlagen und Publikationen der Bissantz & Company GmbH sind geistiges Eigentum von Bissantz & Company oder der Autoren.



## Abstract

*Besonders bei komplexen Planungsanwendungen oder umfangreicheren ETL-Vorgängen erschließen sich manche SQL-Server-Probleme nur bei genauer Betrachtung aller involvierten Faktoren. Ist reines „Logging“ nicht mehr genug oder kommt eine Live-Betrachtung in Frage, sollte statt zu dem Tool „SQL Server Profiler“ zu dem Tool „Extended Events“ gegriffen werden. „Extended Events“ ist seit dem SQL Server 2008 verfügbar und ab dem SQL Server 2012 direkt integriert.*

*In diesem Beitrag zeigen wir, welche Möglichkeiten mit „Extended Events“ zur Verfügung stehen. Dabei erläutern wir anhand von Praxisbeispielen die Vorgehensweise.*

# Probleme im Griff mit Extended Events

## 1 Extended Events

Jeder hat so seine Tools und wenn es um die Nachverfolgung, Nachstellung oder Analyse von komplexen Vorgängen im SQL Server geht, denken viele natürlich zunächst an den SQL Server Profiler. Dieser ist nun seit fast 20 Jahren das grafische Auswertungstool der eventbasierten SQL Server Trace in Microsoft SQL Server.

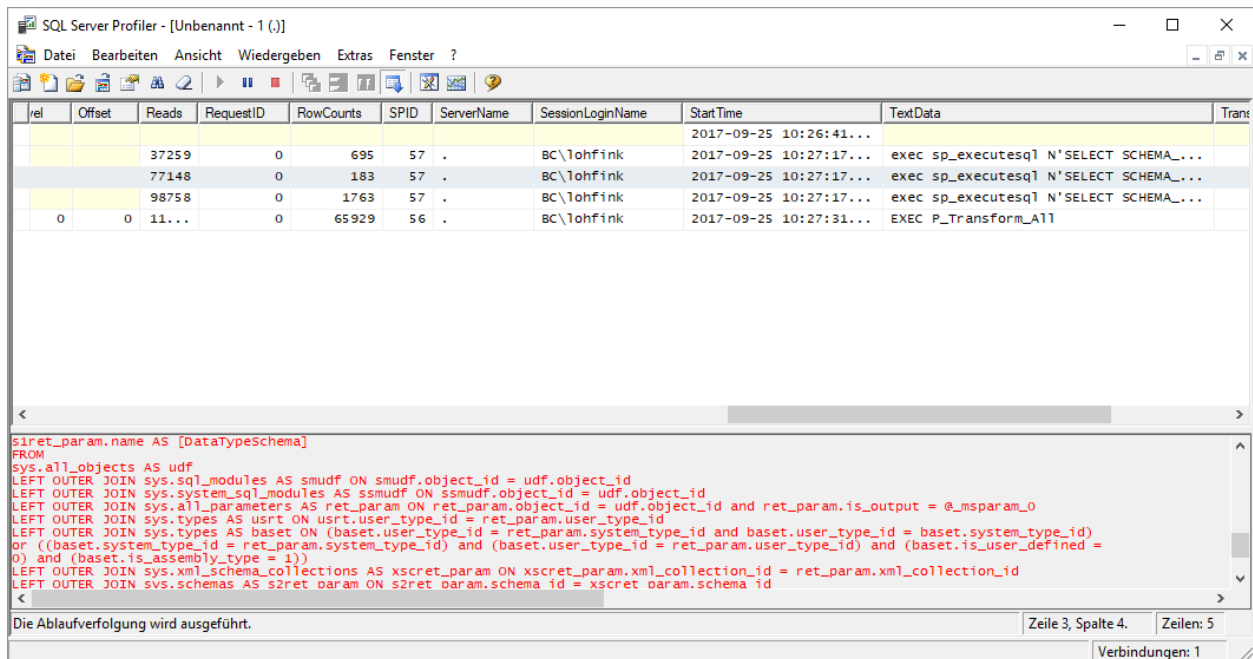


Abbildung 1 Kleines Fenster, großer Inhalt - SQL Server Profiler sieht alles und weiß alles

Seit SQL Server 2008 schlummert jedoch unter der Haube des SQL Servers ein weit mächtigeres Tool als die alte SQL Server Trace: Extended Events

Seit SQL Server 2012 haben wir dazu auch eine grafische Auswertung direkt in [SQL Server Management Studio](#) direkt integriert. Zufälligerweise gilt SQL Server Profiler seit diesem Release auch als „deprecated“.

Der direkte Vorteil von Extended Events gegenüber SQL Server Profiler liegt jedoch in der Behebung einiger Designfehler. So müssen bisher alle Events vom Server an den Client geschickt werden, der den SQL Server Profiler laufen lässt. Dieser Vorgang von extrahieren des Events und anschließendem verschicken etc. kann bei größeren Events deutliche negative Auswirkungen auf die Performance des SQL Servers haben. Bei Extended Events werden diese grundsätzlich zunächst in einen Buffer geladen und dabei können sogar Einstellungen wie: „Wenn wir gerade Performance Probleme haben, können auch Events vergessen werden.“ Berücksichtigt werden. Grundsätzlich sind Aufzeichnungen mit Extended Events also wesentlich verträglicher und können möglicherweise auch im Produktivbetrieb durchgeführt werden.

## 2 Neue Möglichkeiten mit Extended Events

Im Vergleich zum Vorgänger ist es in Extended Events ohne weitere Exporte möglich komplexere Aufzeichnungsbedingungen zu verwenden – wie z.B. „Zeichne nur die ersten x Ereignisse eines Events auf.“ oder „Verwende als Filter eine Bedingung aus meiner ETL-Logik.“ Darüber hinaus können Werte auch ohne Export kumuliert werden, um z.B. die Gesamtanzahl an Lesevorgängen in einer gewissen Situation schnell zu analysieren. Eine weitere beliebte Möglichkeit ist das Auffinden „neuer Maxima“, wobei ein Event nur noch dann aufgezeichnet wird, wenn es z.B. einen Höchstwert in der Ausführungszeit hat.

### 2.1 Kausalitätsverfolgung (Causality tracking)

Während die zuvor genannten Möglichkeiten grundsätzlich nicht neu sind, sondern nun einfach „erleichtert“ werden, da kein Export und anschließende Auswertung per SQL oder Excel notwendig ist, bieten Extended Events jedoch auch eine hervorzuhebende Möglichkeit, die im Vorgänger unmöglich war.

Microsoft nennt diese Fähigkeit Kausalitätsverfolgung und beschreibt damit die logische Gruppierung von Events nach deren Auslösung.

Hier ein Beispiel:

Angenommen zwei User werden aufgezeichnet und planen parallel in einem SQL System Vertriebszahlen. Dabei werden dutzende Prozeduren und Neuberechnungen parallel ausgeführt. Die reine chronologische Darstellung dieser Ereignisse ist durcheinander – okay man könnte noch den Username filtern, aber sobald anwendungsspezifische User ins Spiel kommen oder User-neutrale Vorgänge wie Caching mit aufgezeichnet werden sollen, fällt es sehr schwer zu verstehen „warum“ etwas geschehen ist.

Extended Events kann alle diese Events nach einem Auslöser gruppieren. So können auch parallele Vorgänge oder Vorgänge, die sich von einer klaren Identität lösen nachvollzogen werden.

### 2.2 Default Payloads

Eingangs erwähnt wurde das ungünstige Verhalten von SQL Server Profiler alle Events zum Client, auf dem ebendieser läuft, zu übertragen. Hierzu noch ein paar mehr Worte:

Ein weiteres Problem an der Verarbeitung von Events in SQL Server Profiler ist, dass diese immer vollständig erfasst werden, auch wenn explizit Felder bei der Erstellung der Aufzeichnung nicht ausgewählt wurden. Dennoch muss das darunterliegende SQL Server Trace alle Events vollständig aufzeichnen und anschließend, werden die unerwünschten Spalten herausgefiltert, was zusätzlichen Overhead verursacht.

Bei Extended Events gibt es einen so genannten „Default Payload“ das sind jene Felder, die immer aufgezeichnet werden müssen – sozusagen „physikalisch“ anfallen. Darüber hinaus können die Events dann um weitere Felder angereichert werden. Das können z. B. globale Variablen wie z.B. der Benutzername eines Users sein – oder aber Event spezifische Hilfen, wie z.B. die Auflösung einer `OBJECT_ID` in einen `OBJECT_NAME`.

### 3 Praxisbeispiel

Im Folgenden möchten wir mit Extended Events laufende Prozeduren und T-SQL-Statements überwachen, sowie über Fehler informiert werden.

#### 3.1 Anlegen einer neuen Extended Events Sitzung

Im Vergleich zu SQL Server Profiler erscheinen Extended Events etwas weniger spontan. Während man sich mit SQL Server Profiler jederzeit einfach auf einen Server schalten kann, muss für Extended Events zunächst eine Sitzung angelegt werden.

Dies findet im SQL Server Management Studio im Ordner Verwaltung statt. Hier sind die zu Deutsch „Erweiterten Ereignisse“ zu Hause.

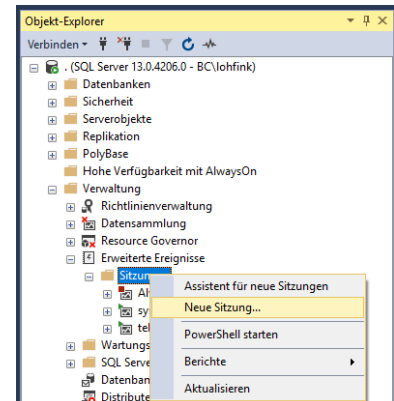


Abbildung 2 Anlegen einer neuen Extended Events Sitzung

#### 3.2 Allgemein

Im Folgenden Dialog wird der Name der Sitzung festgelegt. Des Weiteren lassen sich hier einige wichtige Einstellungen treffen. So kann hier festgelegt werden, ob die Aufzeichnung „permanent“ – also direkt nach dem Start von SQL Server gestartet wird, ob sie im Anschluss an ihre Erstellung gestartet wird und oder ob auch sofort das Live-Logging angezeigt werden soll.

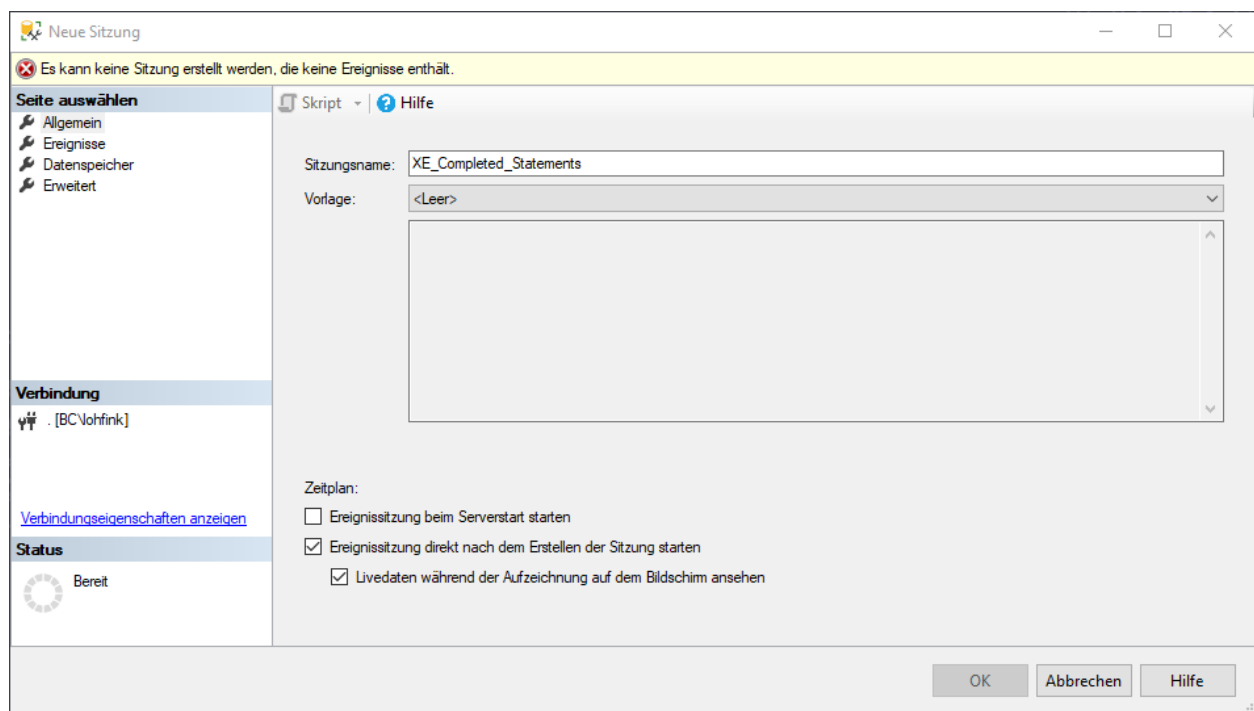


Abbildung 3 Allgemein

### 3.3 Ereignisse

Anschließend wechseln wir im Menü links in den Bereich „Ereignisse“. Hier können jetzt die zu beobachtenden Ereignisse ausgewählt werden. Im Vergleich zum SQL Server Profiler steht hierzu auch eine Suche zu Verfügung.

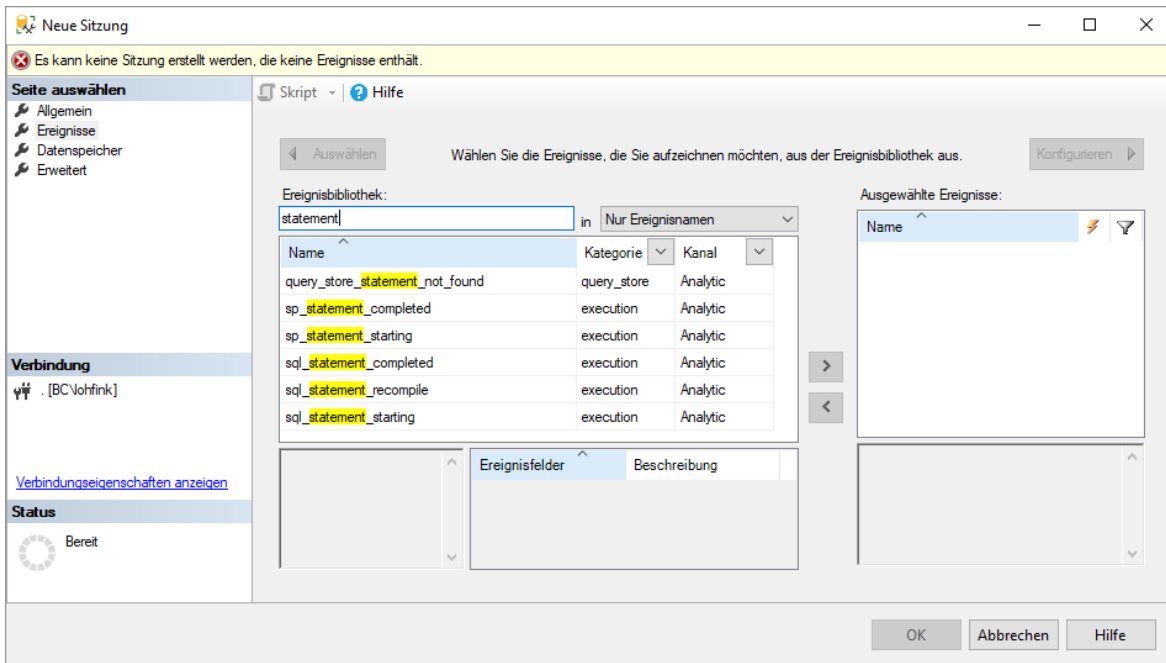


Abbildung 4 Ereignisse

Wir fügen also die Ereignisse `sql_statement_completed`, `sp_statement_completed` und `error_reported` zu unserer Liste hinzu. „`sp_statement_completed`“ steht hierbei für „stored procedure completed“.

Mit einem Klick auf Konfigurieren rechts oben können die ausgewählten Ereignisse wie eingangs erwähnt um zusätzliche Felder erweitert werden oder Filter für bestimmte Werte eingerichtet werden.

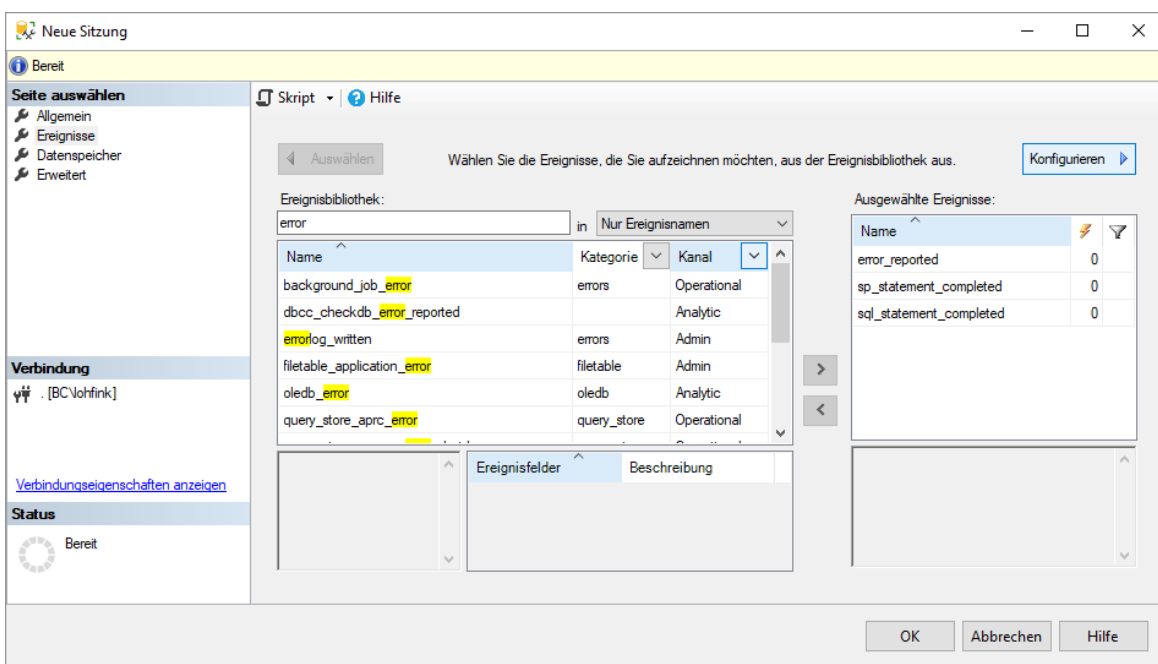


Abbildung 5 Ereignisse auswählen

Im Reiter Ereignisfelder können zusätzliche Felder ausgewählt werden, die Informationen enthalten, die direkt aus dem aufgezeichneten Event ableitbar sind. Die meisten Felder gehören jedoch zum Default Payload und können später nur im UI ausgeblendet werden. Eine Eigenheit dieses Menüs ist, dass in der Liste links (Ausgewählte Ereignisse) mehrere Elemente ausgewählt werden können und die Filter und Felder auf der rechten Seite, dann für alle ausgewählten Ereignisse gültig sind.

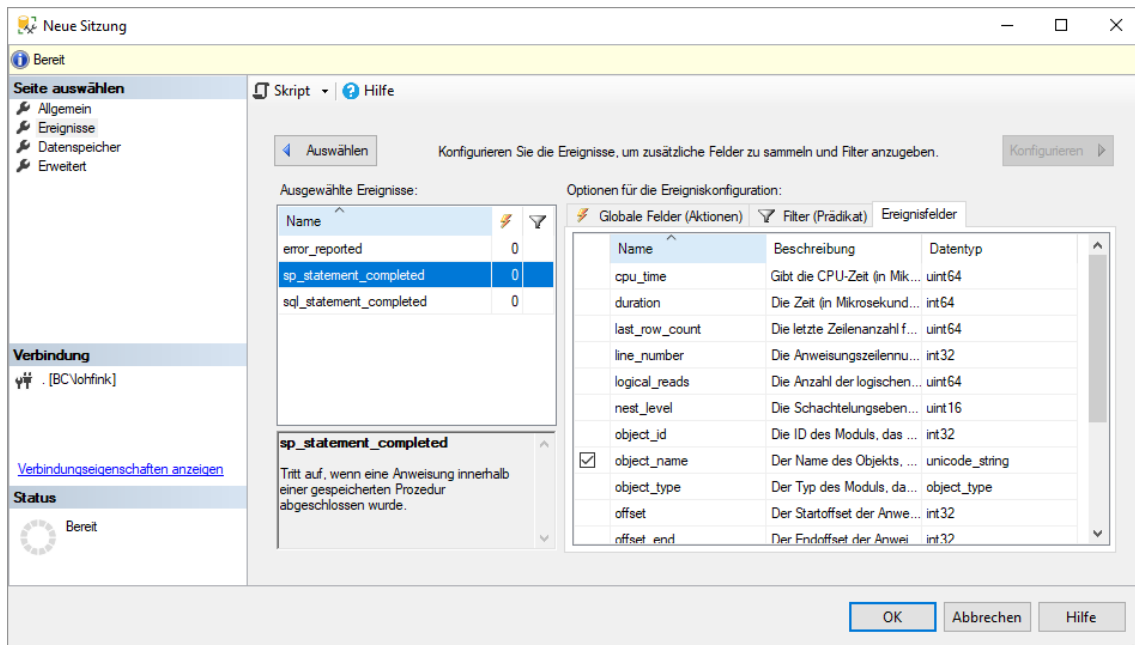


Abbildung 6 Ereignisfelder konfigurieren

Interessant ist das vorallem beim Filter (Prädikat). Eigentlich handelt es sich dabei um eine einfache WHERE Clause und T-SQL ermöglicht hier einen gewaltigen Spielraum, der nicht komplett vom UI abgedeckt werden kann. Es sollte jedoch noch einmal angemerkt werden, dass in Extended Events Filter auch für einzelne Ereignisse ausgewählt werden können, wohingegen SQL Server Profiler nur Filter für alle Ereignisse gemeinsam erlaubt.

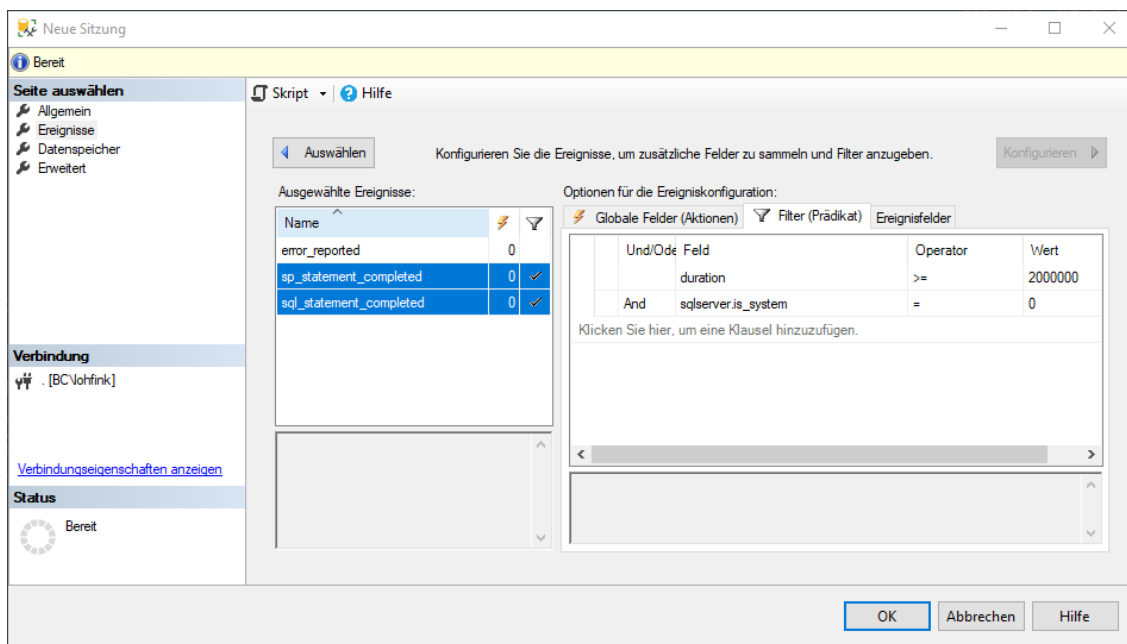


Abbildung 7 Prädikate konfigurieren

Der Reiter „Globale Felder“ ermöglicht das „Anhängen“ von allgemein zur Verfügung stehenden Informationen, wie Session-ID, Username, Verbindungsparameter usw.

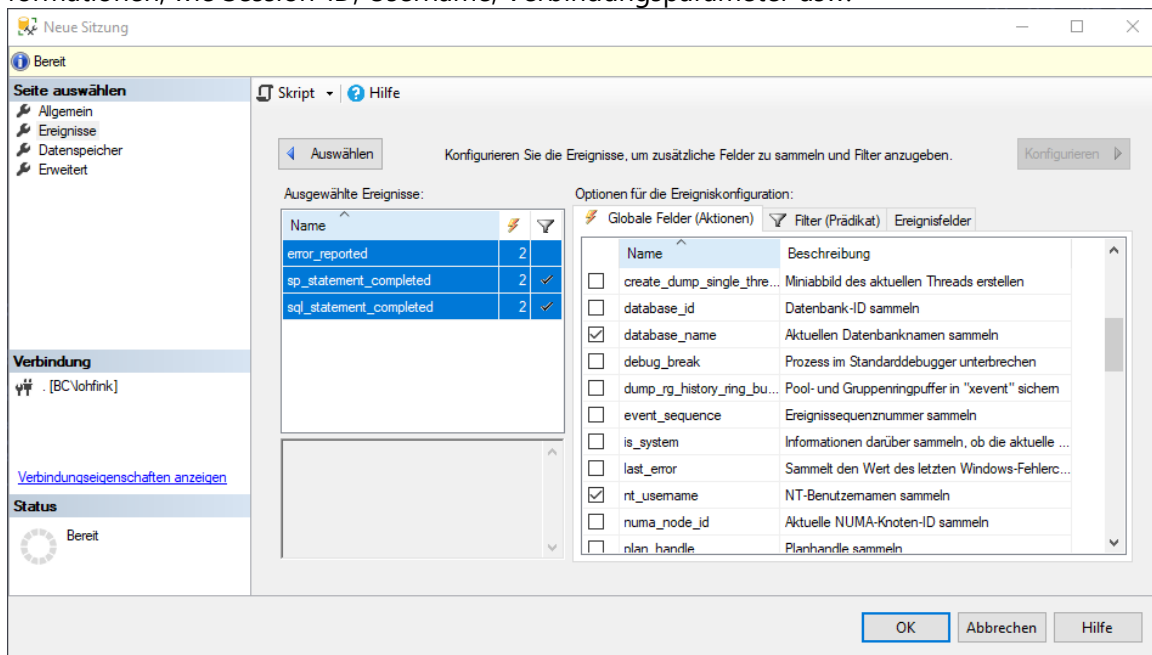


Abbildung 8 Globale Felder konfigurieren

### 3.4 Datenspeicher

In diesem Bereich können mehrere Ziele als Speicherort für die Aufzeichnung ausgewählt werden, von einfachen Dateien, bis hin zu SQL Servers internem [Ring Buffer](#). Auch Log-Rollover ist möglich. In diesem Beispiel konfigurieren wir jedoch nur eine einfache einzelne Datei, da wir keine permanente Nutzung des Ereignisses planen. Grundsätzlich sollten Log-Dateien nicht zu groß werden. Verwenden Sie daher eine Maximalgröße von 512 MB und Dateirollover – also die Erstellung einer neuen Datei sobald die letzte voll ist – falls Sie umfangreichere Logs erwarten.

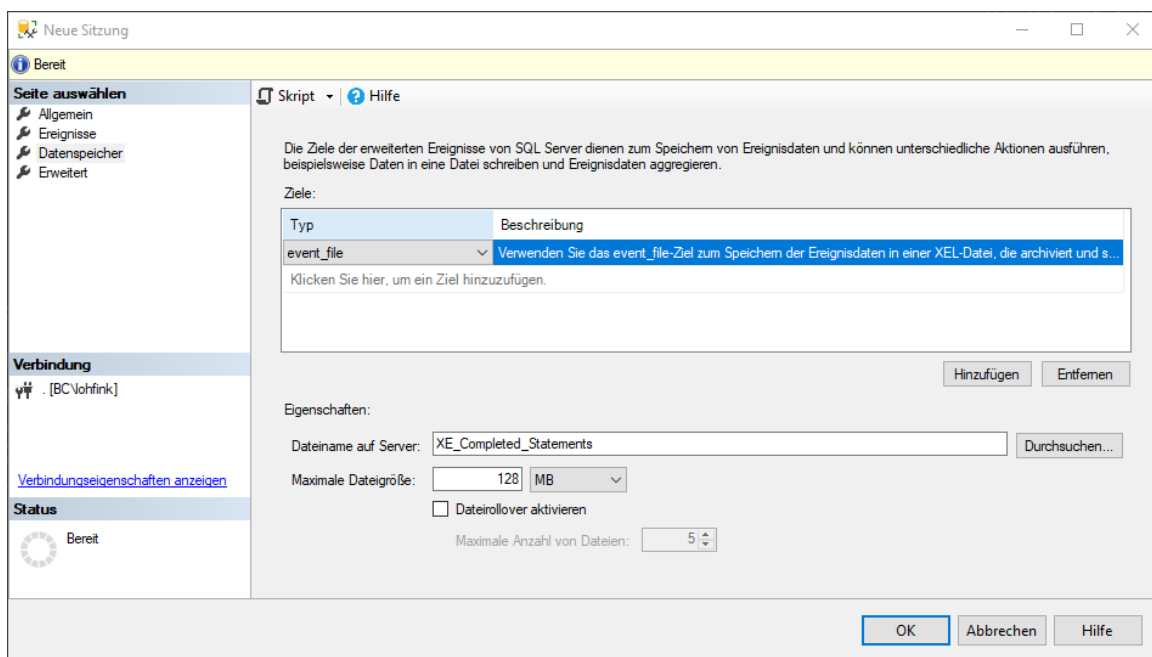


Abbildung 9 Dateispeicher



### 3.5 Erweitert

In diesem Bereich kann der maximal zu verwendende Arbeitsspeicher pro Aufzeichnung eingestellt werden. Um Performanceprobleme wie in SQL Server Profiler zu vermeiden, kann hier eingestellt werden, dass z.B. einzelne Ereignisse auch verloren gehen können, wenn der ausgewählte Arbeitsspeicher voll, aber noch nicht auf z.B. die Festplatte zurückgeschrieben worden ist. Rechnen Sie also mit einer Fülle an Ereignissen, so sollte etwas mehr Arbeitsspeicher und eine schnelle Festplatte als Ziel gewählt werden. Der reservierte Arbeitsspeicher kann auch pro [NUMA Node](#) oder Prozessor ausgewählt werden. Bei NUMA Nodes ist dies von Vorteil, wenn Sie keine Latenzen beim Transport von Node zu Node riskieren möchten.

Als Faustregel können Sie hier 16 MB als maximale Arbeitsspeichergröße angeben. Die 0 in maximale Ereignisgröße bedeutet, dass es keine maximale Ereignisgröße – bzw. nur die sich aus der maximalen Arbeitsspeichergröße ergebende – gibt-

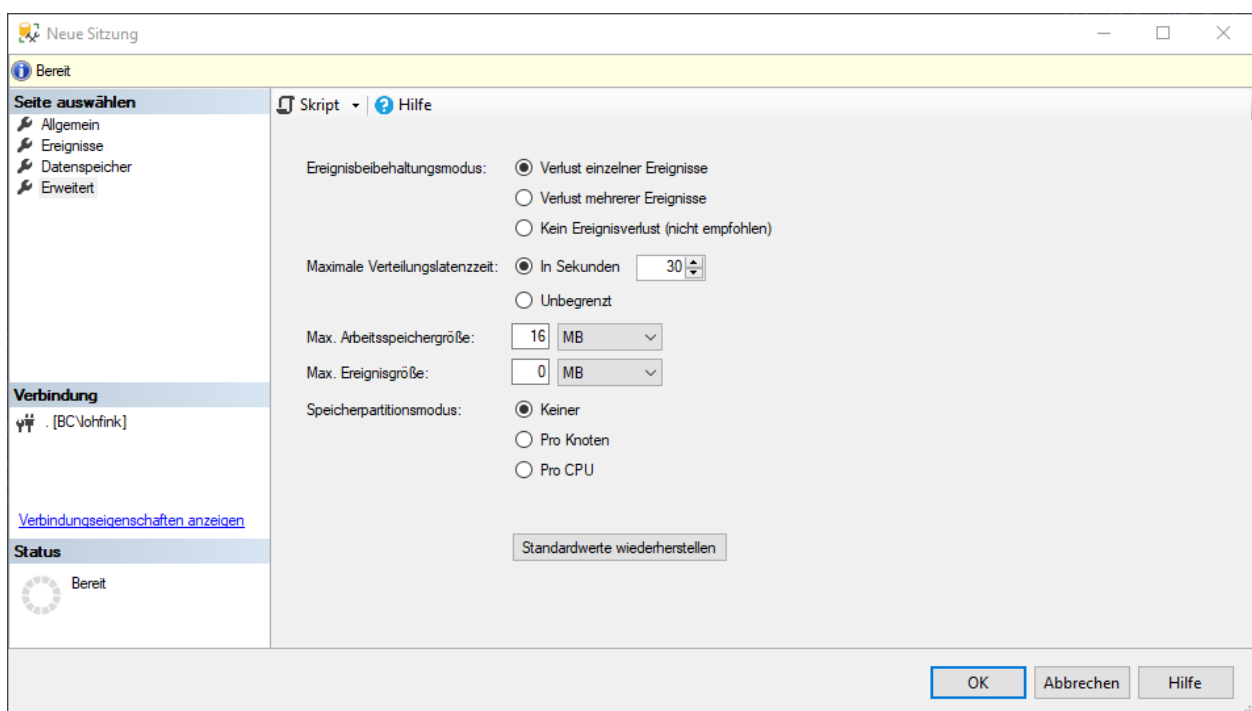


Abbildung 10 Erweitert

### 3.6 Starten der Aufzeichnung

Wenn Sie unter „Allgemein“ die Option „Ereignissitzung direkt nach dem Erstellen starten“ ausgewählt haben, startet die Sitzung nachdem wir OK geklickt haben sofort. Ansonsten genügt ein Rechtsklick auf die Sitzung und die Option „Sitzung starten“. Per Rechtsklick auf die Sitzung kommen wir auch zur Live-Betrachtung, falls wir das nicht ebenfalls unter „Allgemein“ ausgewählt hatten. Daraufhin erscheint die Aufzeichnung, und sieht erstmal etwas langweiliger aus als SQL Server Profiler. Denn in Extended Events werden zunächst immer nur zwei Spalten angezeigt.

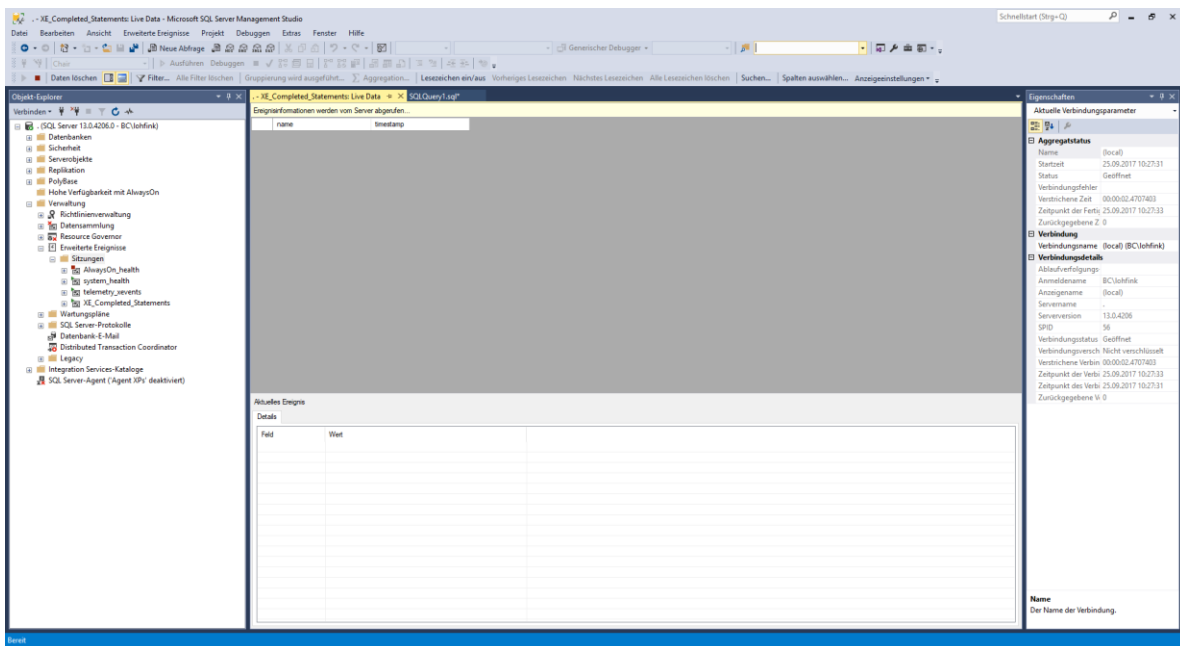


Abbildung 11 Eine gestartete Aufzeichnung, ohne Ereignisse

Mit einem Rechtsklick auf einen beliebigen Spaltenkopf lassen sich auch andere Felder einblenden: Dazu wählen wir „Spalten auswählen“. Nun gleicht das Interface zunächst sehr dem SQL Server Profiler und es lassen sich bereits einige wertvolle Informationen gewinnen!

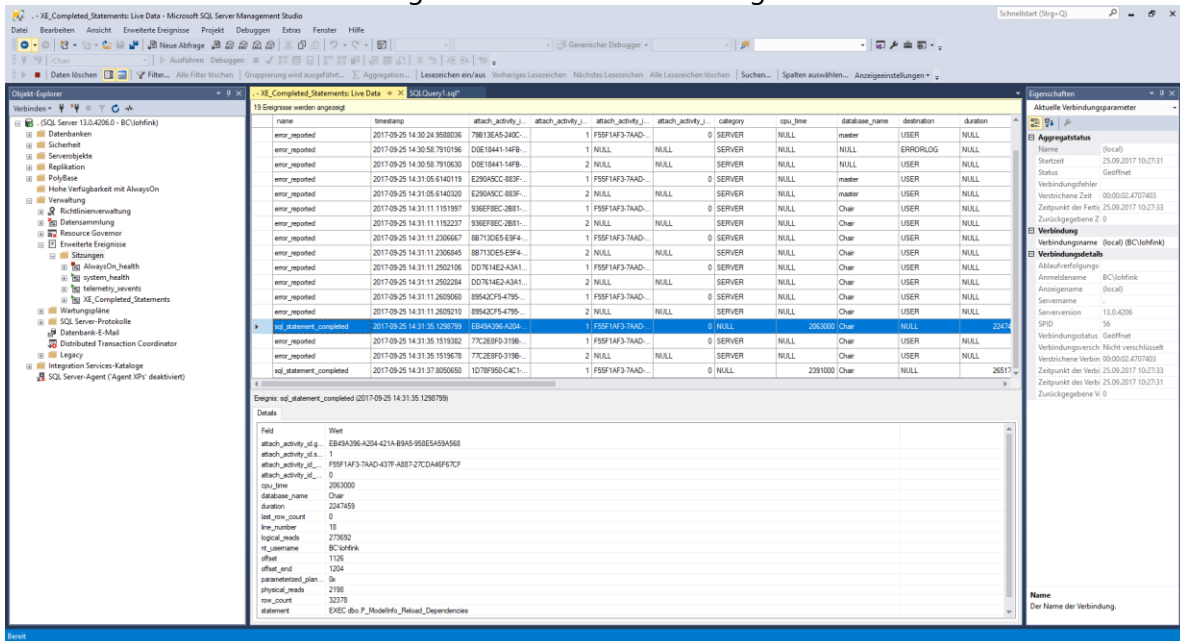


Abbildung 12 Eine laufende Aufzeichnung mit mehreren eingblendeten Spalten

Möchten wir die Aufzeichnung nach dem eingangs erwähnten „kausalen Zusammenhang“ – also der kausalen Kette an Auslösern - analysieren, so müssen wir die Aufzeichnung zunächst anhalten – das geht zum Beispiel mit einem Klick auf den roten Stopp-Button links oben im SQL Server Management Studio Menüband.

Anschließend klicken wir rechts auf die Spalte „attach\_activity\_id\_xfer“ und wählen „nach dieser Spalte gruppieren“ aus. Natürlich könnten wir hier auch nach Benutzername oder anderen Inhalten gruppieren können!

Ein Rechtsklick auf eine numerische Spalte ermöglicht das Berechnen von Aggregationen, wie Summe, Minimum usw.

#### 4 Weitere Möglichkeiten

Mit Extended Events sind viele Dinge möglich, die zuvor mit SQL Server Profiler insbesondere aus Performancesicht nicht zu empfehlen waren. Fortgeschrittene Nutzer sollten sich unbedingt die erweiterten Möglichkeiten der Filter (Prädikate) von Extended Events ansehen, um z.B. eine Filterlogik zu konstruieren, die auf den Zustand der Anwendung Rücksicht nimmt. Des Weiteren ist es auch möglich beim Auftreten von Ereignissen so genannte „Actions“ abzufeuern, um z. B. zusätzliches Logging zu starten.

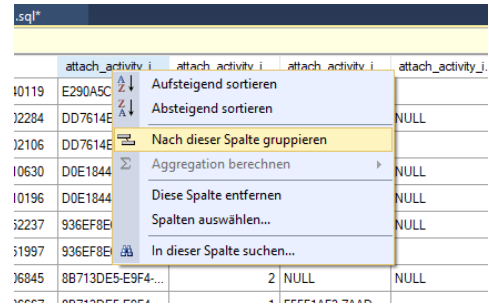


Abbildung 13 Spalten gruppieren

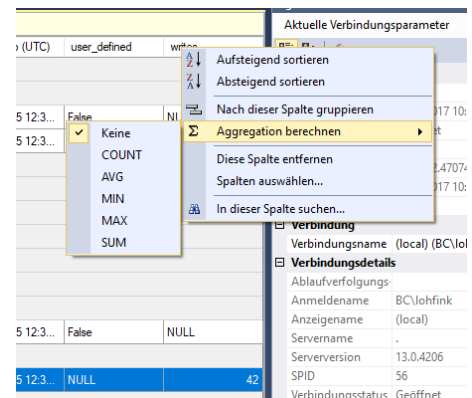


Abbildung 14 Spalten aggregieren