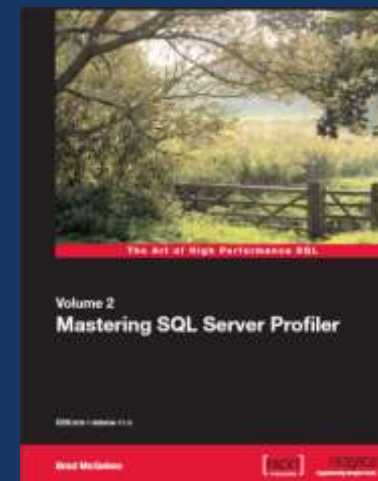




*The World's Largest Community
of SQL Server Professionals*

Getting the Most Out of the SQL Server 2005/2008 Profiler

Brad M McGehee
Director of DBA Education
[Red Gate Software](http://www.redgate.com)
bradmckehee@hotmail.com



My Assumptions About You

- You have been a SQL Server DBA or developer for one year or longer.
- You have used Profiler before, and understand the basics.
- You want to learn more about how to use Profiler in order to help you better do your job.

My Goals for Today

- Make you aware of the power of Profiler.
- Provide you some examples of how to use Profiler to help you indentify and troubleshoot some common, and uncommon, problems.
- To get you *excited* about using Profiler so that when you go back to work, you will *make* the time to learn more about Profiler, and that you will *begin to use it regularly* to help you become a better DBA.

Profiler Has Enormous Potential

- Profiler is one of the most powerful, but often underutilized tools in the DBA toolkit.
- It's not a perfect tool, and in some cases, there are better tools available to resolve specific problems.
- SQL Server 2008 Profiler offers 180 different events it can monitor. This doesn't include AS events.
- For each event, it can also collect up to 64 different types of data.

Ways to Use Profiler

- Monitor the activity of:
 - SQL Server Database Engine
 - Analysis Services
- Monitor and identify performance-related problems with front-end applications by capturing and viewing the T-SQL sent from the application to SQL Server.
- Perform query analysis using graphical execution plans
- Monitor for and identify SQL Server errors and warnings
- Capture data for analysis with the Database Engine Tuning Wizard
- Audit user activity
- Capture and replay traces of activity to replicate problems
- Debug T-SQL code and stored procedures
- Correlate Profiler trace events with System Monitor performance counters to identify what event(s) are hogging server resources

What About SQL Trace?

- SQL Trace is the engine that is used by the Profiler GUI to capture and store trace data.
- If you prefer, you can avoid using the Profiler GUI and use SQL Trace exclusively.
- SQL Trace is configured and managed using T-SQL system stored procedures.
- SQL Trace uses less overhead than the Profiler GUI, and is somewhat more flexible. It is also harder to set up and use compared to the GUI.

Our Focus for Today

- Because we can't cover everything in a single session, we will focus on how to use Profiler to:
 - Identify Long Duration Queries
 - Identify Blocked Processes
 - Identify Deadlocks
 - Identify Memory Warnings
 - Correlate Profiler and Performance Monitor Data

These are all examples of how you might use Profiler, and many DBAs may choose to perform the above tasks differently from what I am showing you.

How to Identify Long Duration Queries

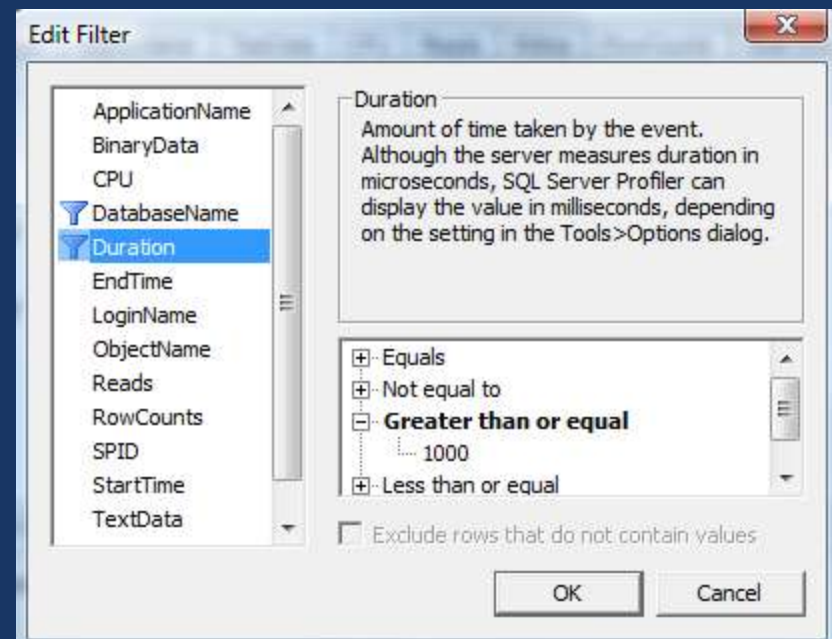
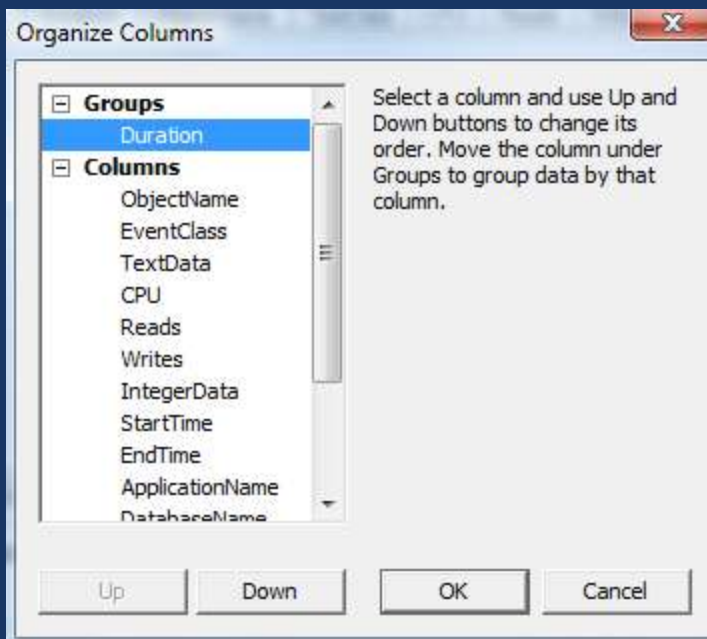
- Queries can cause many different kinds of performance problems.
- One option is to identify those queries that are taking the longest to run, in addition to gathering useful information that can be used to troubleshoot why they are performing slowly.
- Note: This method does not consider how often a query executes.

Common Events

- The **RPC:Completed** event fires after a stored procedure is executed as a remote procedure call. There is a **RPC:Starting** that can be used.
- The **SQL:BatchCompleted** event fires *after* a stored procedure is executed and completed using the EXECUTE statement, *plus* whenever a T-SQL batch is executed.
- A **SQL:BatchStarting** event fires *before* a stored procedure is executed using the EXECUTE statement, *plus* whenever a T-SQL batch is executed. Useful to see when a query starts.
- The **SP:StmtCompleted** event tells us when a statement within a stored procedure has completed.
- The **Showplan XML** event displays the estimated graphical execution plan of a query.

How to Identify Long Duration Queries

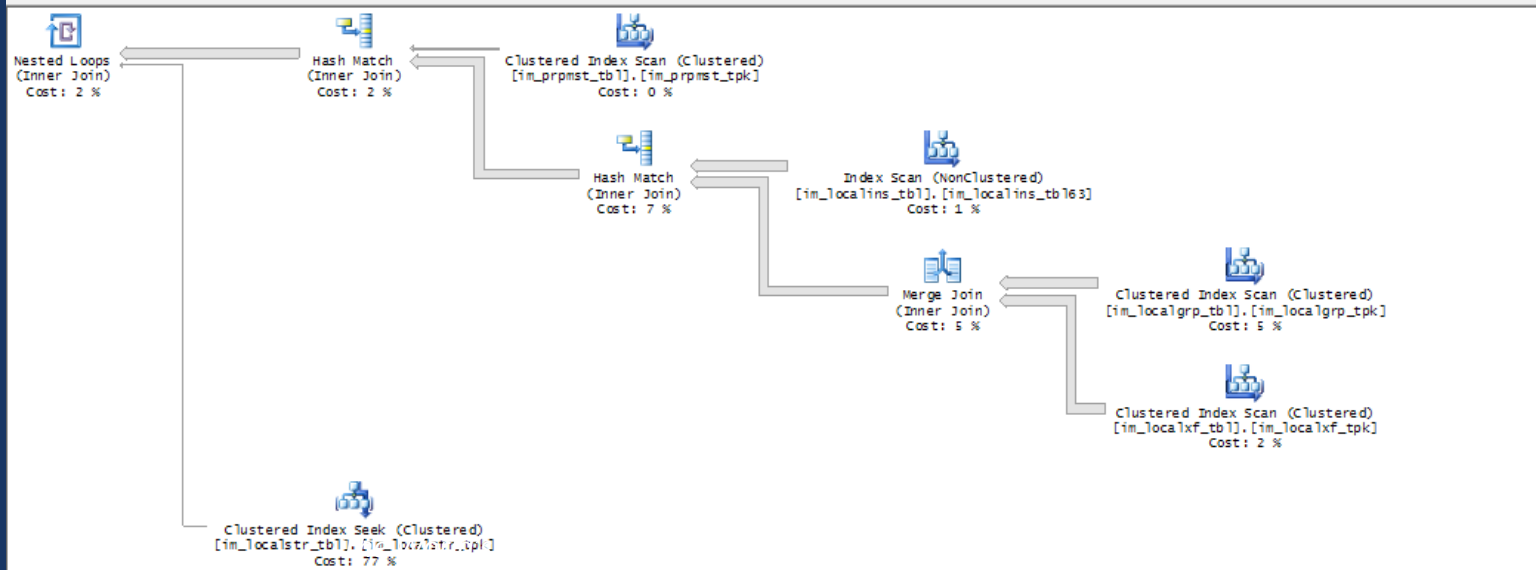
Events	Duration	ObjectName	TextData	CPU	Reads	Writes	IntegerData	StartTime	EndTime	Application
Performance										
<input checked="" type="checkbox"/> Showplan XML		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Stored Procedures										
<input checked="" type="checkbox"/> RPC:Completed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SP:StmtCompleted	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TSQL										
<input checked="" type="checkbox"/> SQL:BatchCompleted	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SQL:BatchStarting			<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>



How to Identify Long Duration Queries

Duration	ObjectName	EventClass	TextData	CPU	Reads	Writes	IntegerData
(13)							
1769 (1)							
3723 (1)							
3724 (1)							
9055 (1)							
9056 (1)							
11837 (2)							
11837	ADGSP_EN_Instructions	SP:StmtCompleted	SELECT dbo.im_localstr_tbl.im_l...	3479	110379	0	582828
11837		SQL:BatchCompleted	EXEC dbo.ADGSP_EN_Instructions ;	3479	110382	0	

Duration	ObjectName	EventClass	TextData	CPU	Reads	Writes	IntegerData
		SQL:BatchStarting	EXEC dbo.ADGSP_EN_Instructions ; --...				
	ADGSP_EN_Instructions	Showplan XML	<ShowPlanXML xmlns="http://schemas....				449877
10646	ADGSP_EN_Instructions	SP:StmtCompleted	SELECT dbo.im_localstr_tbl.im_local...	3759	1859482	0	582828
10715		SQL:BatchCompleted	EXEC dbo.ADGSP_EN_Instructions ; --...	3822	1859603	0	



How to Identify Blocked Processes #1

- Blocked processes can significantly reduce server performance and upset users.
- Identifying the causes of a blocked process is not always easy.
- Profiler makes it somewhat easier to identify blocking processes, helping to identify the code that is causing the problem.

How to Identify Blocked Processes #2

- The Blocked Process event is not turned on by default, you must turn it on using these steps:
 1. Turn on: `sp_configure 'blocked process threshold'`
 2. Set an appropriate time frame in seconds
 3. Configure Profiler and run the trace to produce report
 4. Turn off: `sp_configure 'blocked process threshold'`

How to Identify Blocked Processes

```
sp_configure 'show advanced options', 1 ;
```

```
GO
```

```
RECONFIGURE ;
```

```
GO
```

```
sp_configure 'blocked process threshold', 5 ;
```

```
GO
```

```
RECONFIGURE ;
```

```
GO
```

How to Identify Blocked Processes

General | Events Selection

Review selected events and event columns that are being traced. Selection cannot be changed while tracing is active.

Events	DatabaseID	Duration	EndTime	EventSequence	IndexID	IsSystem	LoginSid	Mode
Errors and Warnings								
<input checked="" type="checkbox"/> Blocked process report	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

How to Identify Blocked Processes

EventClass	DatabaseID	Duration	EndTime	EventSequence	IndexID	IsSystem	LoginSid	Mod
Trace Start								
Blocked process report	5	9850	2008-08-25 16:43:10...	407732	0	1	0X01	8.
Blocked process report	5	14899	2008-08-25 16:43:15...	407733	0	1	0X01	8.

```

<blocked-process-report>
<blocked-process>
  <process id="process41fe40" taskpriority="0" logused="0" waitresource="OBJECT: 5:754101727:0 " waittime="14899"
ownerId="1694874" transactionname="user_transaction" lasttranstarted="2008-08-25T16:43:01.067" XDES="0x582fbe0" lockMode="IX"
schedulerid="2" kpid="5836" status="suspended" spid="66" sbid="0" ecid="0" priority="0" trancount="2"
lastbatchstarted="2008-08-25T16:43:00.973" lastbatchcompleted="2008-08-25T16:43:00.973" clientapp="Microsoft SQL Server
Management Studio - Query" hostname="BIGISLAND" hostpid="6912" loginname="BigIsland\BradM" isolationlevel="read committed (2)"
xactid="1694874" currentdb="5" lockTimeout="4294967295" clientoption1="671090784" clientoption2="390200">
  <executionStack>
    <frame line="4" stmtstart="88" sqlhandle="0x02000000a5155c2cb85ee89a270f471ceb11c6da7aff9a46"/>
  </executionStack>
  <inputbuf>
--This code is being blocked
BEGIN TRAN
UPDATE sales.[SalesOrderHeader]
SET [DueDate] = GETDATE()
  </inputbuf>
</process>
</blocked-process>
<blocking-process>
  <process status="suspended" waittime="17687" spid="61" sbid="0" ecid="0" priority="0" trancount="1"
lastbatchstarted="2008-08-25T16:42:56.803" lastbatchcompleted="2008-08-25T16:42:56.787" clientapp="Microsoft SQL Server
Management Studio - Query" hostname="BIGISLAND" hostpid="6912" loginname="BigIsland\BradM" isolationlevel="read committed (2)"
xactid="1694842" currentdb="5" lockTimeout="4294967295" clientoption1="671090784" clientoption2="390200">
  <executionStack>
    <frame line="6" stmtstart="196" sqlhandle="0x0200000041a4ce18bcdade97ede7fd52f83ea77b512be8ebb"/>
  </executionStack>
  <inputbuf>
--Begin Lock of Table
BEGIN TRAN
SELECT *
FROM Sales.SalesOrderHeader WITH ( HOLDLOCK )
WAITFOR DELAY &apos;00:05:00&apos;
  </inputbuf>
</process>
</blocking-process>
</blocked-process-report>

```

How to Identify Deadlocking Issues

- A deadlock occurs when two or more spids block each other with each spid having a lock on a resource that another spid is trying to simultaneously lock.
- Because neither spid can complete its task, nor can either resource be released until each spid completes its task, a deadlock occurs.
- In most cases, the spid that has used the fewest resources up to the point of the deadlock will be automatically killed by a SQL Server lock monitor thread, allowing the other spid to complete.
- Troubleshooting deadlocking issues can often be difficult, but Profiler can help.

How to Identify Deadlocking Issues

Events	TextData	ApplicationName	DatabaseName	ServerName	Duration	CPU	Reads	Writes
Locks								
<input checked="" type="checkbox"/> Deadlock graph	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>				
<input checked="" type="checkbox"/> Lock:Deadlock	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
<input checked="" type="checkbox"/> Lock:Deadlock Chain	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Stored Procedures								
<input checked="" type="checkbox"/> RPC:Completed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SP:StmtCompleted	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TSQL								
<input checked="" type="checkbox"/> SQL:BatchCompleted	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SQL:BatchStarting	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				

Organize Columns

Select a column and use Up and Down buttons to change its order. Move the column under Groups to group data by that column.

Groups

- Columns**

EventClass
TextData
ApplicationName
DatabaseName
ServerName
Duration
CPU
Reads
Writes
EventSequence
LoginName
SPID

Up Down OK Cancel

Edit Filter

ApplicationName
BigintData1
BinaryData
CPU
DatabaseName
Duration
EndTime
Error
EventSubClass
HostName
IntegerData
IntegerData2
LoginName

DatabaseName
Name of the database in which the statement of the user is running.

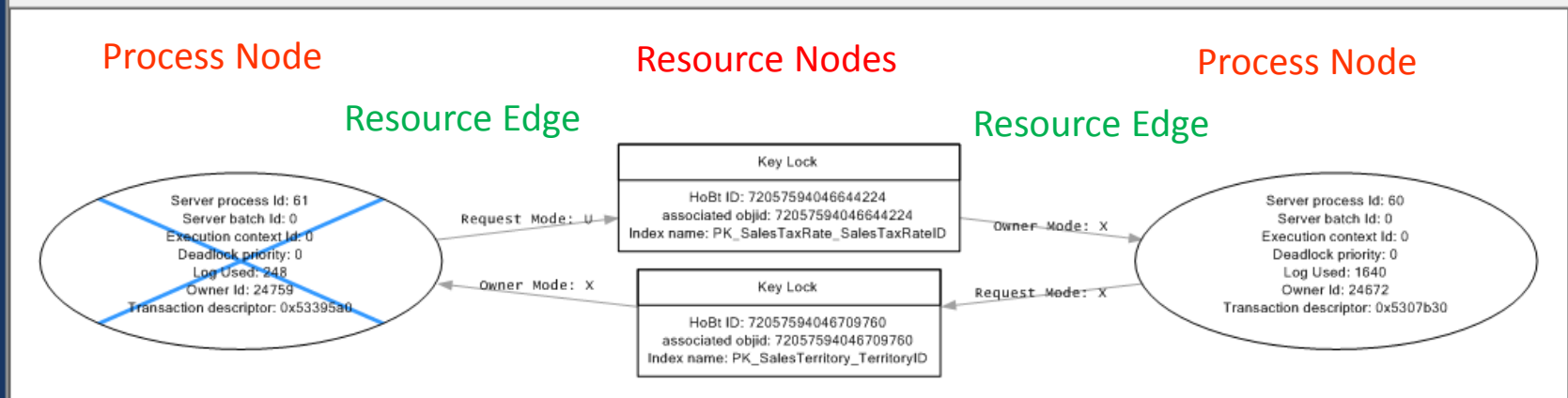
Like
 AdventureWorks
 Not like

Exclude rows that do not contain values

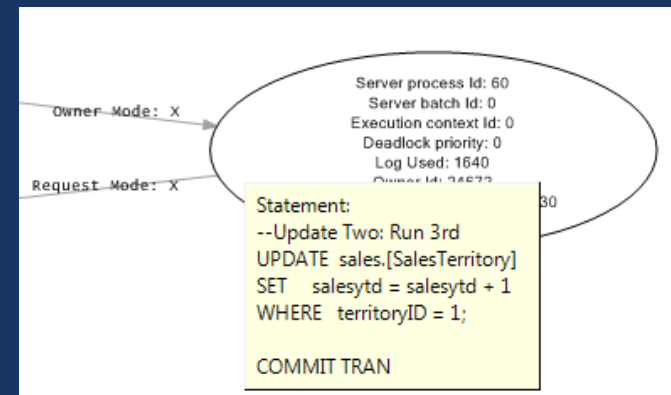
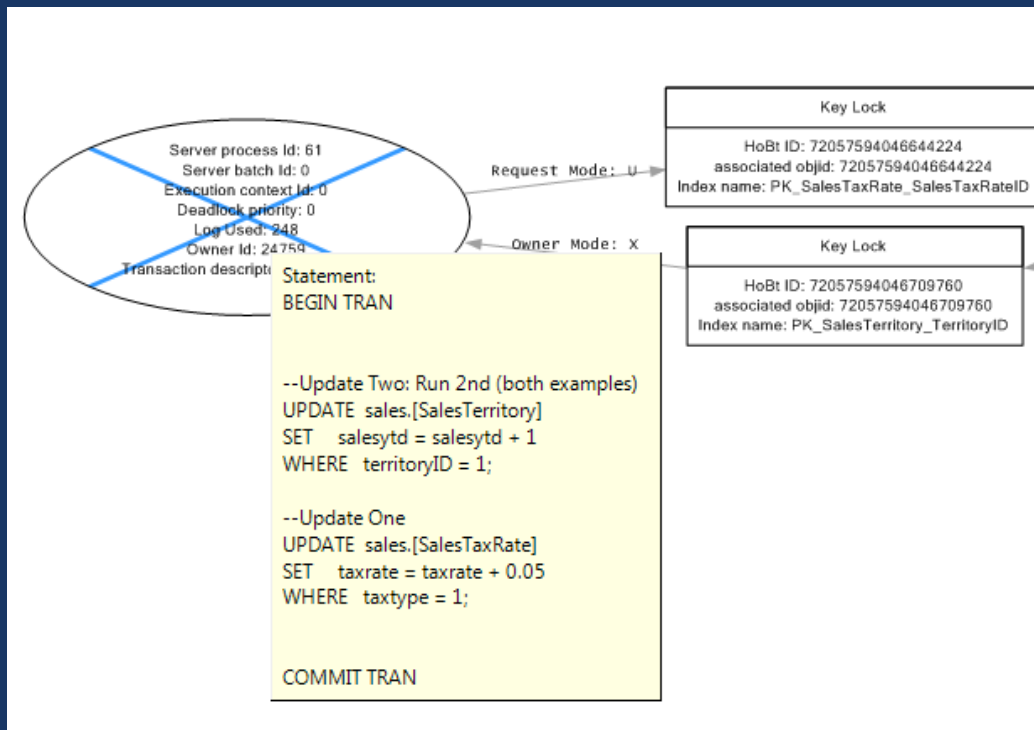
OK Cancel

How to Identify Deadlocking Issues

EventClass	TextData	ApplicationName	DatabaseName	ServerName	Duration	CPU	Reads
SQL:BatchCompleted	USE [Adventureworks]	Microsoft...	Adventureworks	BIGISLAND\SQL2008	0	0	0
SP:StmtCompleted	SELECT StatMan([SCO]) FROM (SELECT TOP 100 ...	Microsoft...	Adventureworks	BIGISLAND\SQL2008	0	0	2
SP:StmtCompleted		Microsoft...	Adventureworks	BIGISLAND\SQL2008	29	0	26
SQL:BatchStarting	BEGIN TRAN --Update One: Run 1st UPDATE...	Microsoft...	Adventureworks	BIGISLAND\SQL2008			
SQL:BatchCompleted	BEGIN TRAN --Update One: Run 1st UPDATE...	Microsoft...	Adventureworks	BIGISLAND\SQL2008	128	0	228
SQL:BatchCompleted	USE [Adventureworks]	Microsoft...	Adventureworks	BIGISLAND\SQL2008	0	0	0
SQL:BatchStarting	BEGIN TRAN --Update Two: Run 2nd (both...	Microsoft...	Adventureworks	BIGISLAND\SQL2008			
SQL:BatchStarting	--Update Two: Run 3rd UPDATE sales.[Sales...	Microsoft...	Adventureworks	BIGISLAND\SQL2008			
Lock:Deadlock Chain	Deadlock Chain SPID = 61 (010086470766) ...		Adventureworks	BIGISLAND\SQL2008			
Lock:Deadlock Chain	Deadlock Chain SPID = 60 (010086470766) ...		Adventureworks	BIGISLAND\SQL2008			
Lock:Deadlock	(010086470766)	Microsoft...	Adventureworks	BIGISLAND\SQL2008	12074		
Deadlock graph	<deadlock-list> <deadlock victim="process...			BIGISLAND\SQL2008			
SQL:BatchCompleted	BEGIN TRAN --Update Two: Run 2nd (both...	Microsoft...	Adventureworks	BIGISLAND\SQL2008	12111	0	305
SQL:BatchCompleted	--Update Two: Run 3rd UPDATE sales.[Sales...	Microsoft...	Adventureworks	BIGISLAND\SQL2008	1405	0	4
Trace Stop							



How to Identify Deadlocking Issues

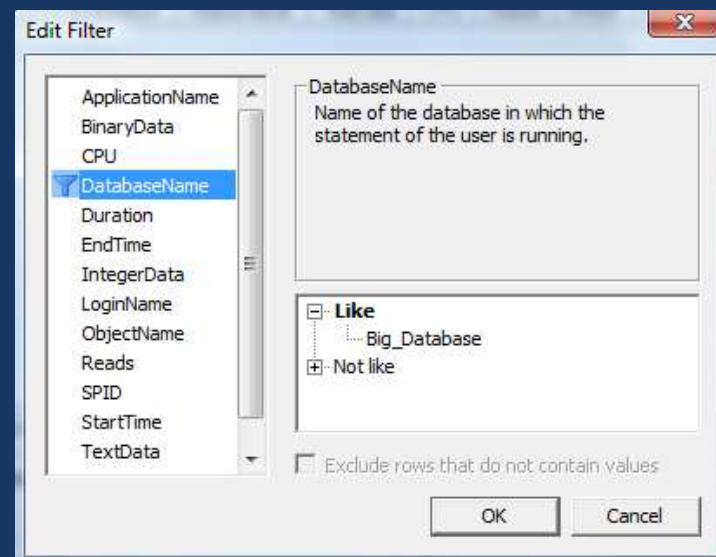
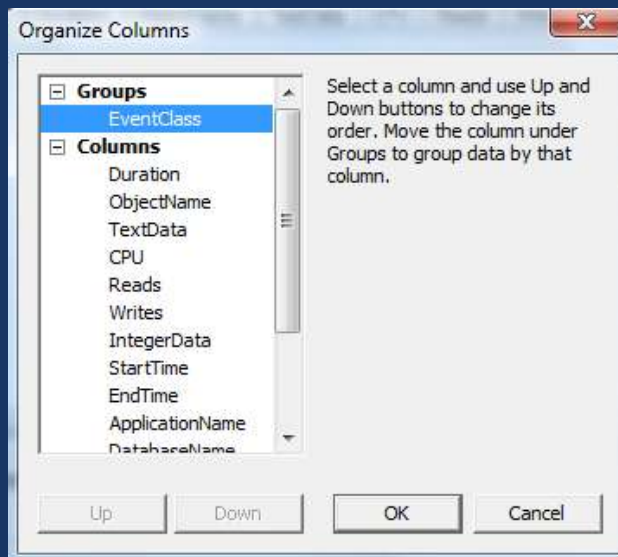


How to Identify Memory Warnings

- There are three Profiler events that can help us understand different types of memory pressures in SQL Server. They include:
 - **Execution Warnings**: Event fires whenever a query has to wait one second or more before it is allocated the memory it needs to execute.
 - **Sort Warnings**: Event fires whenever SQL Server is unable to complete a sort entirely in memory, having to write to tempdb.
 - **Server Memory Change**: Event fires when the memory of a SQL Server instance increases or decreases by either 1mb, or by 5% of the maximum server memory allocated to it, whichever is greater

How to Identify Memory Warnings

Events	Duration	ObjectName	TextData	CPU	Reads	Writes	IntegerData	StartTime	EndTime	ApplicationName
<input checked="" type="checkbox"/> Execution Warnings	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>		
<input checked="" type="checkbox"/> Sort Warnings								<input checked="" type="checkbox"/>		
Performance										
<input checked="" type="checkbox"/> Showplan XML		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Server										
<input checked="" type="checkbox"/> Server Memory Change							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Stored Procedures										
<input checked="" type="checkbox"/> RPC:Completed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/> SP:StmtCompleted	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
TSQL										
<input checked="" type="checkbox"/> SQL:BatchCompleted	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/> SQL:BatchStarting			<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>		



How to Identify Memory Warnings

SQL:BatchStarting			SELECT * FROM ap_vchr_tbl w...		
Showplan XML		Dynamic...	<ShowPlanXML xmlns="http://schemas....		
Server Memory Change					
Sort Warnings					
SQL:BatchCompleted	15441		SELECT * FROM ap_vchr_tbl w...	2620	18073
SQL:BatchStarting			SELECT * FROM ap_vchr_tb...		
Showplan XML		Dynamic...	<ShowPlanXML xmlns="http://schemas....		
Sort Warnings					
SQL:BatchCompleted	11788		SELECT * FROM ap_vchr_tb...	2231	16441

How to Identify Excessive Auto Stats Activity

- If the AUTO UPDATE STATISTICS options is turned on for a database, then based on an internal algorithm, SQL Server will automatically update a table's index or column statistics.
- In most cases, the automatic updating of statistics is good thing
- But there are cases when it can cause problems.

How to Identify Excessive Auto Stats Activity

- Potential problems caused by Auto Stats:
 - Statistics aren't updated often enough, resulting in poor execution plans.
 - Statistics are updated too often, using unnecessary resources; causing recompiles.
 - Statistics updating may run at inappropriate times, causing random performance hits.
 - Depending if synchronous or asynchronous updating is done, can contribute to inconsistent query times.

How to Identify Excessive Auto Stats Activity

- The Auto Stats event tells you:
 - How often auto stats are being run
 - How long is it taking to run auto stats
 - Which databases and objects are incurring the most auto stats events
 - What kind of auto stats event is it: synchronous or asynchronous

How to Identify Excessive Auto Stats Activity

Events	EventSubClass	Duration	DatabaseName	TextData	ApplicationName	EndTime	EventSequence	...
Performance								
<input checked="" type="checkbox"/> Auto Stats	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Edit Filter

ApplicationName
DatabaseName
Duration
EndTime
EventSubClass
IndexID
IntegerData
IsSystem
LoginName
ObjectID
SPID
StartTime
TextData

DatabaseName
Name of the database in which the statement of the user is running.

Like
 Demo
 Not like

Exclude rows that do not contain values

OK Cancel

Organize Columns

Groups
Columns
EventClass
EventSubClass
Duration
DatabaseName
TextData
ApplicationName
EndTime
EventSequence
IndexID
IntegerData
IsSystem
LoginName

Select a column and use Up and Down buttons to change its order. Move the column under Groups to group data by that column.

Up Down OK Cancel

How to Identify Excessive Auto Stats Activity

EventClass	EventSubClass	Duration	DatabaseName	TextData
Trace Start				
Auto Stats	1	440	Demo	Updated: dbo.DemoTable.DemoStatistics
Auto Stats	1	582	Demo	Updated: dbo.DemoTable.DemoTableIndex
Auto Stats	1	388	Demo	Updated: dbo.DemoTable.DemoStatistics
Auto Stats	1	396	Demo	Updated: dbo.DemoTable.DemoTableIndex
Trace Stop				

Updated: dbo.DemoTable.DemoStatistics

EventSubClass	Description
1	Statistics created/updated synchronously; TextData column indicates which statistics and whether they were created or updated.
2	Asynchronous statistics update; job queued.
3	Asynchronous statistics update; job starting.
4	Asynchronous statistics update; job completed.

How to Identify Excessive Table/Index Scans

- Index and table scans are a fact of life.
- In some cases, they are a good thing.
- But in many cases, excessive index and table scans may indicate potential performance problems that need to be researched.
- Profiler can identify when index and table scans occur.

How to Identify Excessive Table/Index Scans

Events	ObjectID	Duration	ObjectName	TextData	CPU	Reads	Writes	IntegerData	StartTime	EndTime
Performance										
<input checked="" type="checkbox"/> Showplan XML	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Scans										
<input checked="" type="checkbox"/> Scan:Started	<input checked="" type="checkbox"/>								<input checked="" type="checkbox"/>	
Stored Procedures										
<input checked="" type="checkbox"/> RPC:Completed		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SP:StmtCompleted	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TSQL										
<input checked="" type="checkbox"/> SQL:BatchCompleted		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SQL:BatchStarting				<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>	

Edit Filter

ApplicationName
 BinaryData
 ClientProcessID
 CPU
 DatabaseID
 DatabaseName
 Duration
 EndTime
 Error
 GroupID
 HostName
 IndexID
 IntegerData

ApplicationName
 Name of the client application that created the connection to SQL Server. This column is populated with the values passed by the application rather than the displayed name of the program.

Like
 Not like
 SQL Server Profiler - 45a1d69f-8547-4c

Exclude rows that do not contain values

OK Cancel

Organize Columns

Groups
 EventClass
 Columns
 ObjectID
 Duration
 ObjectName
 TextData
 CPU
 Reads
 Writes
 IntegerData
 StartTime
 EndTime
 ApplicationName

Select a column and use Up and Down buttons to change its order. Move the column under Groups to group data by that column.

Up Down OK Cancel

How to Identify Excessive Table/Index Scans

EventClass	ObjectID	Duration	ObjectName	TextData	CPU	Reads	Writes
SQL:BatchStarting				SELECT text1 FROM dbo.[DemoT...			
Showplan XML	645178831		Dynamic...	<ShowPlanXML xmlns="http://schemas....			
Scan:Started	2137058649						
SQL:BatchCompleted		14		SELECT text1 FROM dbo.[DemoT...	31	854	0
SQL:BatchStarting				SELECT text2 FROM dbo.[DemoT...			
Showplan XML	51840727		Dynamic...	<ShowPlanXML xmlns="http://schemas....			
Scan:Started	2137058649						
SQL:BatchCompleted		1174		SELECT text2 FROM dbo.[DemoT...	16	854	0

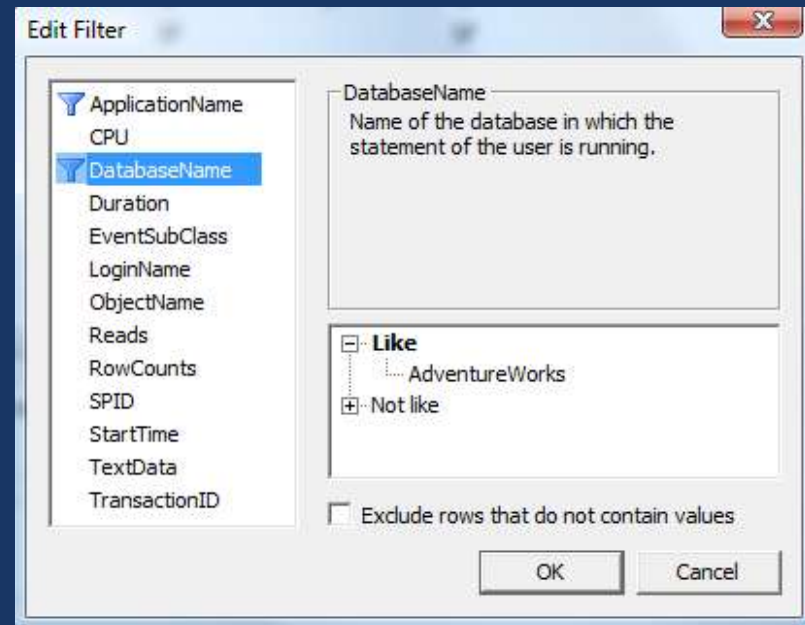
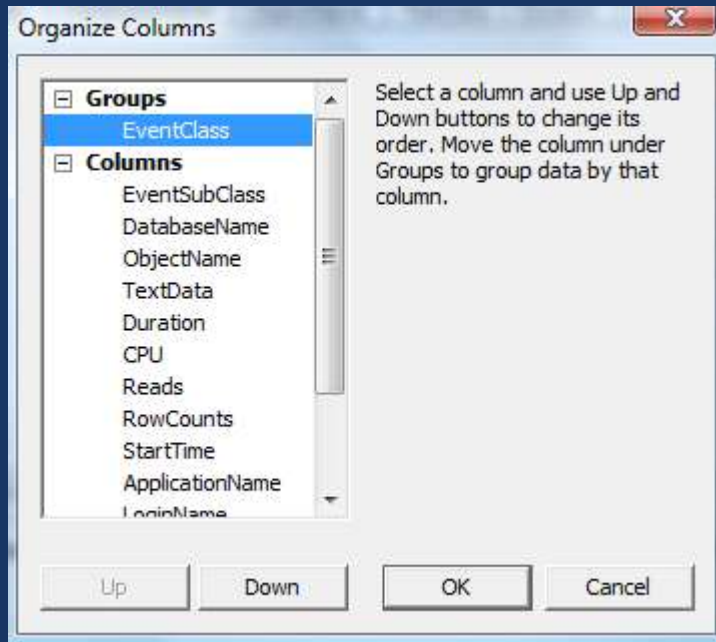
```
SELECT text1
FROM dbo.[DemoTableTwo]
WHERE TEXT1 = '55TKIL4GPVR2VZ73DOQW9EZ0ZGTP7NL6G8B3JWU'
go
```

How to Identify Excessive SP Compilations

- Generally, it is ideal to minimize the number of compilations SQL Server performs. This helps reduce resource overhead.
- In some cases, recompilations are a good thing. Recompilations, depending on the reason, can help ensure that a query is executing with an optimal execution plan.
- But in other cases, recompilations can become excessive, wasting server resources.
- In this case, we need to identify the excessive recompilations and reduce them.
- In order to reduce them, we must first identify them.

How to Identify Excessive SP Compilations

Events	EventSubClass	DatabaseName	ObjectName	TextData	Duration	CPU	Reads	RowCounts	StartTime
TSQL									
<input checked="" type="checkbox"/> SQL:StmtCompleted		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SQL:StmtRecompile	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SQL:StmtStarting		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>



How to Identify Excessive SP Compilations

EventClass	EventSubClass	DatabaseName	ObjectName	TextData
SQL:StmtRecompile (8)				
SQL:StmtRecompile	3 - Deferred compile	Adventure...	DemoProc	SELECT * FROM #t1
SQL:StmtRecompile	3 - Deferred compile	Adventure...	RowCoun...	INSERT INTO #t1
SQL:StmtRecompile	3 - Deferred compile	Adventure...	RowCoun...	SELECT a
SQL:StmtRecompile	2 - Statistics changed	Adventure...	RowCoun...	SELECT a
SQL:StmtRecompile	1 - Schema changed	Adventure...		INSERT INTO t1 VALUES (
SQL:StmtRecompile	12 - Parameterized plan flushed	Adventure...		INSERT INTO t1 VALUES (
SQL:StmtRecompile	1 - Schema changed	Adventure...		INSERT INTO t2 VALUES (
SQL:StmtRecompile	12 - Parameterized plan flushed	Adventure...		INSERT INTO t2 VALUES (
SQL:StmtStarting (23)				
Trace Start (1)				


```
INSERT INTO t1
VALUES (
    'Trailing blank ',
    'Trailing blank ',
    0x00ee00
);
```

How to Identify Excessive SP Compilations

EventClass	EventSubClass	DatabaseName	ObjectName	TextData
SQL:StmtRecompile	2 - Statistics changed	Adventureworks	RowCoun...	SELECT a
SQL:StmtStarting		Adventureworks		SET ANSI_PADDING ON ;
SQL:StmtStarting		Adventureworks		CREATE TABLE t1 (
SQL:StmtStarting		Adventureworks		INSERT INTO t1 VALUES (
SQL:StmtRecompile	1 - Schema changed	Adventureworks		INSERT INTO t1 VALUES (
SQL:StmtStarting		Adventureworks		INSERT INTO t1 VALUES (
SQL:StmtStarting		Adventureworks		INSERT INTO t1 VALUES (
SQL:StmtRecompile	12 - Parameterized plan flushed	Adventureworks		INSERT INTO t1 VALUES (
SQL:StmtStarting		Adventureworks		INSERT INTO t1 VALUES (

Describes the cause of the recompilation:

- 1 = Schema changed
- 2 = Statistics changed
- 3 = Deferred compile
- 4 = Set option changed
- 5 = Temp table changed
- 6 = Remote rowset changed
- 7 = For Browse permissions changed
- 8 = Query notification environment changed
- 9 = Partition view changed
- 10 = Cursor options changed
- 11 = Option (recompile) requested

How to Identify Plan Cache Activity

- In most cases, a T-SQL query should be compiled once, its execution plan cached, then reused.
- To find out if a cached plan is being reused or not, we can use these events:
 - CacheHit: Plan found
 - CacheMiss: Plan not found
 - CacheInsert: Plan added
 - CacheRemove: Plan removed

How to Identify SP Cache Activity

General | Events Selection

Review selected events and event columns to trace. To see a complete list, select the "Show all events" and "Show all columns" options.

Events	Duration	ObjectName	TextData	CPU	Reads	Writes	StartTime	EndTime	ApplicationName
Stored Procedures									
<input checked="" type="checkbox"/> RPC:Completed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SP:CacheHit		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SP:CacheInsert			<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SP:CacheMiss		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SP:CacheRemove			<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
TSQL									
<input checked="" type="checkbox"/> SQL:BatchCompleted	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SQL:BatchStarting			<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Edit Filter

ApplicationName
 BinaryData
 ClientProcessID
 CPU
 DatabaseID
 DatabaseName
 Duration
 EndTime
 EventSubClass
 GroupID
 HostName
 IntegerData
 IsSystem

ApplicationName
 Name of the client application that created the connection to SQL Server. This column is populated with the values passed by the application rather than the displayed name of the program.

Like
 Not like
 SQL Server Profiler - 45a1d69f-8547-4c

Exclude rows that do not contain values

OK Cancel

Organize Columns

Groups
 EventClass

Columns
 Duration
 ObjectName
 TextData
 CPU
 Reads
 Writes
 IntegerData
 StartTime
 EndTime
 ApplicationName
 DatabaseName

Select a column and use Up and Down buttons to change its order. Move the column under Groups to group data by that column.

Up Down OK Cancel

How to Identify SP Cache Activity

EventClass	Duration	ObjectName	TextData
Trace Start			
SP:CacheMiss			DBCC FREEPROCCACHE
SQL:BatchStarting			DBCC FREEPROCCACHE
SP:CacheRemove			Entire Procedure Cache Flushed
SQL:BatchCompleted	1		DBCC FREEPROCCACHE
SP:CacheMiss			EXEC dbo.demo_cache ; ...
SQL:BatchStarting			EXEC dbo.demo_cache ; ...
SP:CacheInsert			demo_cache
SP:CacheHit		demo_cache	
SQL:BatchCompleted	4253		EXEC dbo.demo_cache ; ...
SP:CacheInsert			SELECT TOP 50000 ...
SQL:BatchStarting			SELECT TOP 50000 ...
SQL:BatchCompleted	1828		SELECT TOP 50000 ...
SP:CacheHit			SELECT TOP 50000 ...
SQL:BatchStarting			SELECT TOP 50000 ...
SQL:BatchCompleted	1711		SELECT TOP 50000 ...
Trace Stop			

How to Correlate a SQL Server Profiler Trace with Performance Monitor Counter Data

- SQL Server Profiler allows you to correlate Profiler Trace Data with Performance Monitor Performance Counter Data.
- This helps you to better understand the relationship between SQL Server events and server resource utilization.

How to Correlate a SQL Server Profiler Trace with Performance Monitor Counter Data

- Overview of steps
 - First, launch Performance Monitor, and create a performance log file with appropriate counters. Save to a file.
 - Next, start Profiler, and create a trace file with appropriate events, data columns, and filters. Save to a file.
 - Collect data with both tools until all data is collected.
 - Stop and exit both tools.
 - Restart Profiler, load saved Profiler trace.
 - Import System Monitor data file.
 - Analyze Results

How to Correlate a SQL Server Profiler Trace with Performance Monitor Counter Data

- Demo

Profiler Best Practices

- For expensive traces, or for busy servers, consider using SQL Trace instead of using the Profiler GUI.
- Only include those events and data columns you need.
- Avoid including events that occur often.
- Avoid collecting duplicate data.
- Only run Profiler for as long as needed to collect the data you need, during times when the problem is likely to occur.
- Use Filters to reduce how much data is collected and saved.
- If running the Profiler GUI, run it from a computer other than the one you are tracing.
- Store data directly to disk files, not to a database.

Take Aways

- Profiler is a powerful tool, that is not used often enough by DBAs and developers.
- Now that you have seen some of the power of Profiler, hopefully you will take some time and learn more about Profiler.
- The learning curve is steep, but the effort is worthwhile.

Q&A

- Please ask your questions clearly and loudly.
- If you don't get your questions answered now, see me after the session, or e-mail me.

Find Out More

Free E-Books:

www.sqlservercentral.com/Books

Check these out:

www.SQLServerCentral.com

www.Simple-Talk.com

Contact me at:

bradmcgehee@hotmail.com

Blogs:

www.bradmcgehee.com

www.twitter.com/bradmcgehee

[Click Here for a free 14-day trial of the Red Gate SQL Server Toolbelt](#)

