# [Automatic Tuning Monitoring and Diagnostics](http://www.sqlservercentral.com/blogs/sqlrnnr/2019/01/04/automatic-tuning-monitoring-and-diagnostics/)

[SQLRNNR](http://www.sqlservercentral.com/blogs/sqlrnnr/)

Posted on 4 January 2019

FROM: <http://www.sqlservercentral.com/blogs/sqlrnnr/2019/01/04/automatic-tuning-monitoring-and-diagnostics/?utm_source=SSC&utm_medium=pubemail>

[](http://jasonbrimhall.info/wp-content/uploads/2018/12/day11_v2.png)

Cool new toys/tools have been made available to the data professional. Among these tools are query data store and [automatic tuning](https://docs.microsoft.com/en-us/sql/relational-databases/automatic-tuning/automatic-tuning?view=sql-server-2017). These two tools actually go hand in hand and work pretty nicely together.

With most new tools, there is usually some sort of instruction manual along with a section on how to troubleshoot the tool. In addition to the manual, you usually have some sort of guide as to whether or not the tool is working within desired specifications or not.

Thanks to Extended Events (XE), we have access to a guide of sorts that will help us better understand if our shiny new tool is operating as desired.

**Operationally Sound**

XE provides a handful of events to help us in evaluating the usage of Automatic Tuning in SQL Server. To find these events, we can simply issue a query such as the following.

select xo.name, xo.object\_type, xo.description

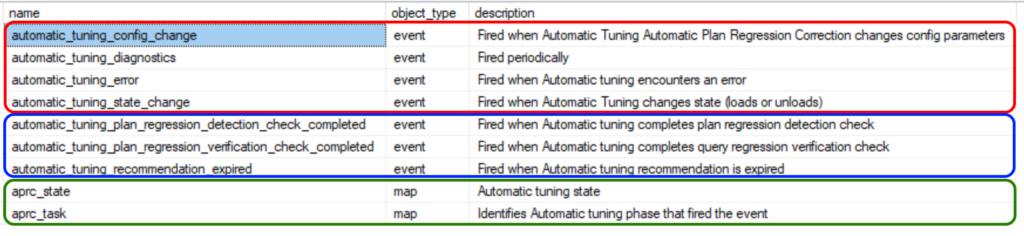
From sys.dm\_xe\_objects xo

where xo.name like '%autom%tun%'

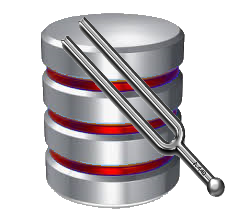
or xo.description like '%autom%tun%'

order by xo.object\_type;

When executed, this query will provide a result set similar to the following.

[](http://jasonbrimhall.info/wp-content/uploads/2019/01/autotune_events.png)

I have grouped the results from this query into three sets. In the red set, I have four events that are useful in the diagnostics and monitoring of automatic tuning. These events show errors, diagnostic (and performance) data, configuration changes and state changes.

[](http://jasonbrimhall.info/wp-content/uploads/2019/01/db_red_highlights_tuning.png)For instance, the state change event will fire when automatic tuning is enabled and will also fire when the database is started (assuming the session is running). The automatic\_tuning\_diagnostics event fires roughly every 30 minutes on my server to gather performance and diagnostic data that can help me understand how well the feature is performing for my workload in each database.

Highlighted in the green section is a couple of maps that show the various values for the current phase or state of the automatic tuning for each database. One can view these different values with the following query.

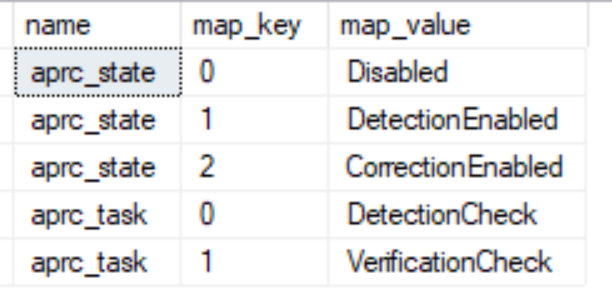
select mv.name,mv.map\_key,mv.map\_value

From sys.dm\_xe\_map\_values mv

where mv.name in ('aprc\_state','aprc\_task')

order by mv.name, mv.map\_key;

This query yields these results.

[](http://jasonbrimhall.info/wp-content/uploads/2019/01/autotune_maps.png)We will see those values in use in the events in a session shortly.

We have seen some of the events and some of the maps at a very quick glance. That said, it is a good time to pull it all together and create a session.

USE master;

GO

-- Create the Event Session

IF EXISTS ( SELECT \*

FROM sys.server\_event\_sessions

WHERE name = 'AutoTuneMonitor' )

DROP EVENT SESSION AutoTuneMonitor

ON SERVER;

GO

EXECUTE xp\_create\_subdir 'C:\Database\XE';

GO

CREATE EVENT SESSION [AutoTuneMonitor] ON SERVER

ADD EVENT qds.automatic\_tuning\_config\_change(

ACTION ( sqlserver.database\_id,

sqlserver.nt\_username, sqlserver.sql\_text, sqlserver.username,sqlserver.session\_nt\_username,

sqlserver.client\_app\_name, sqlserver.session\_id, sqlserver.client\_hostname)

)

,ADD EVENT qds.automatic\_tuning\_diagnostics(

ACTION ( sqlserver.database\_id,

sqlserver.nt\_username, sqlserver.sql\_text, sqlserver.username,sqlserver.session\_nt\_username,

sqlserver.client\_app\_name, sqlserver.session\_id, sqlserver.client\_hostname)

)

,ADD EVENT qds.automatic\_tuning\_state\_change(

ACTION ( sqlserver.database\_id,

sqlserver.nt\_username, sqlserver.sql\_text, sqlserver.username,sqlserver.session\_nt\_username,

sqlserver.client\_app\_name, sqlserver.session\_id, sqlserver.client\_hostname)

)

,ADD EVENT qds.automatic\_tuning\_error(

ACTION ( sqlserver.database\_id,

sqlserver.nt\_username, sqlserver.sql\_text, sqlserver.username,sqlserver.session\_nt\_username,

sqlserver.client\_app\_name, sqlserver.session\_id, sqlserver.client\_hostname)

)

ADD TARGET package0.event\_file(SET filename=N'C:\Database\XE\AutoTuneMonitor.xel')

,ADD TARGET package0.ring\_buffer

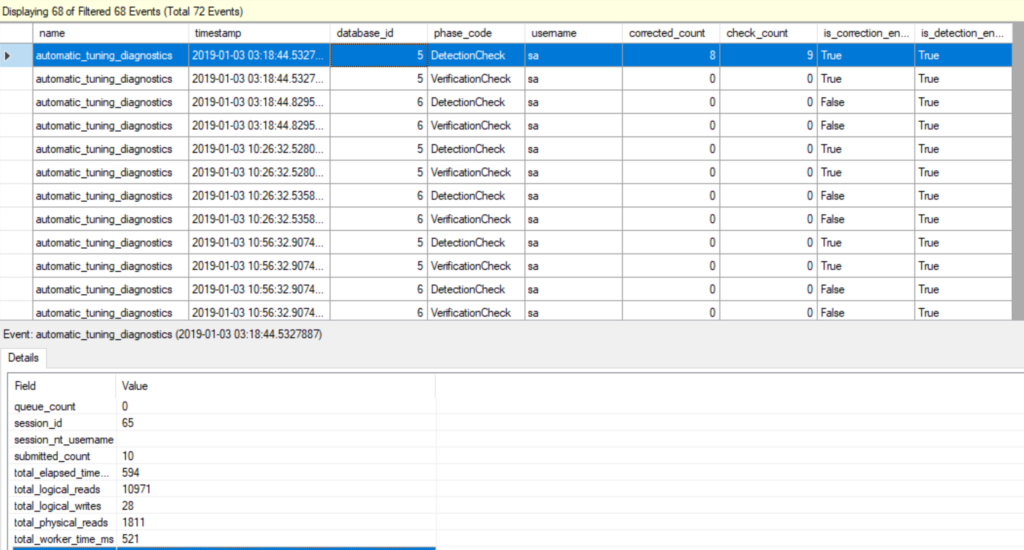
WITH (MAX\_DISPATCH\_LATENCY=5 SECONDS,TRACK\_CAUSALITY=ON,STARTUP\_STATE=OFF);

GO

ALTER EVENT SESSION [AutoTuneMonitor] ON SERVER STATE = START;

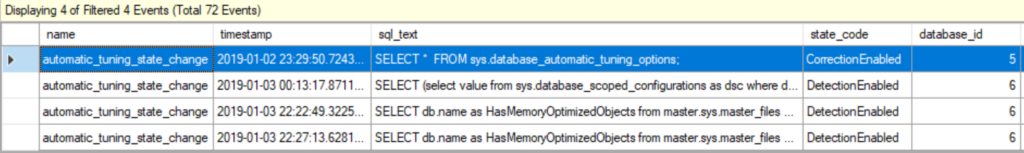
GO

Seeing as this session won’t produce any results without Query data store being enabled and automatic tuning being configured for a database, I have set all of that up in a demo database and have some fresh results to display.

[](http://jasonbrimhall.info/wp-content/uploads/2019/01/autotune_xemonitor2.png)

Here I show an example of the output filtered for just the diagnostics event. Note the phase\_code shows some of those map values previously discussed. I can also see that roughly every 30 minutes each database undergoes a diagnostics check.

Now, looking at another event in that same session, I can see the following.

[](http://jasonbrimhall.info/wp-content/uploads/2019/01/autotune_xemonitor.png)The state\_code in this event payload demonstrates more values from the maps previously discussed (CorrectionEnabled and DetectionEnabled). In this case, the automatic\_tuning\_state\_change fired a few times for database 6 because that database was intentionally taken offline and set back online to test the event.

[](http://jasonbrimhall.info/wp-content/uploads/2018/03/xe_flysuperhero.png)The use of these particular events in this session is very lightweight. I don’t have a predicate configured for any of the events because I wanted to trap everything. Of course, the number of events can increase with an increased load and usage scenarios on different servers.

**The Wrap**

Automatic tuning can be a pretty sharp tool in your tool-belt on your way to becoming that rock-star DBA. As you start to sharpen your skills with this tool, you will need to have some usage and diagnostic information at your fingertips to ensure everything is running steady. This event session is able to provide that diagnostic information and keep you on top of the automatic tuning engine.

For more uses of Extended Events, I recommend my series of articles designed to help you [learn XE little by little](http://jasonbrimhall.info/2015/09/08/learning-extended-events-in-60-days/).

Interested in seeing the **power** of XE over Profiler? [Check this one out](http://jasonbrimhall.info/2017/12/22/an-identity-crisis-is-it-profiler-or-is-it-extended-events/)!

This has been the eleventh article in the **2018 “12 Days of Christmas”** series. For a full listing of the articles, [visit this page](http://bit.ly/2DVEgOO).

http://jasonbrimhall.info/?feed-stats-post-id=5141

**Related Posts:**

* [Dynamics AX Event Session](http://jasonbrimhall.info/2018/01/02/dynamics-ax-event-session/) January 2, 2018
* [Quick and Easy XE for Azure DB](http://jasonbrimhall.info/2018/12/24/quick-and-easy-xe-for-azure-db/) December 24, 2018
* [Monitor Database Offline Events](http://jasonbrimhall.info/2018/11/15/monitor-database-offline-events/) November 15, 2018
* [Checking your Memory with XE](http://jasonbrimhall.info/2019/01/02/checking-your-memory-with-xe/) January 2, 2019

[Checking the Health of your CLR](http://jasonbrimhall.info/2018/01/16/checking-the-health-of-your-clr/) January 16, 2018