**Automated collection of SQL Server database connections for monitoring**

|  |  |
| --- | --- |
|  | By: [Svetlana Golovko](http://www.mssqltips.com/sqlserverauthor/94/svetlana-golovko/)  FROM: <http://www.mssqltips.com/sqlservertip/3193/automated-collection-of-sql-server-database-connections-for-monitoring/> |

**Problem**

We have quite a large number of databases and user connections to our SQL Server and we suspect that one application does not close connections properly. We need an automated way to monitor when this happens in order to contact developers to fix this issue.

**Solution**

In order to collect connection information, we will setup a SQL Server Agent alert that will execute a job every time there is more than N number of connections. We will use 50 connections in our example to simplify the test. The job then will populate a table with information about applications and connections.

**Create a table to store monitoring data**

First, we will create the table that will be populated by the job:

CREATE TABLE perf\_warehouse.dbo.\_demo\_sessions\_alert(

[host\_name] nvarchar(128) NULL,

[program\_name] nvarchar(128) NULL,

login\_name nvarchar(128) NULL,

num\_sessions int NULL,

capture\_time datetime NULL

) ON [PRIMARY]

GO

**Create the SQL Server Agent Job**

Create the SQL Server job with the following script as a step:

INSERT INTO perf\_warehouse.dbo.\_demo\_sessions\_alert

SELECT [host\_name],

[program\_name],

login\_name,

count(c.session\_id ) num\_sessions,

getdate()

FROM sys.dm\_exec\_connections c JOIN

sys.dm\_exec\_sessions s on c.session\_id = s.session\_id

GROUP BY host\_name, program\_name, login\_name ORDER BY 4 DESC

You can capture all connections or only the top N ("SELECT TOP 10 [host\_name],...").

Here is the complete script that you can use to create the job:

BEGIN TRANSACTION

DECLARE @ReturnCode INT

SELECT @ReturnCode = 0

IF NOT EXISTS (SELECT name FROM msdb.dbo.syscategories

WHERE name=N'DBA' AND category\_class=1)

BEGIN

EXEC @ReturnCode = msdb.dbo.sp\_add\_category @class=N'JOB', @type=N'LOCAL', @name=N'DBA'

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

END

DECLARE @jobId BINARY(16)

EXEC @ReturnCode = msdb.dbo.sp\_add\_job @job\_name=N'Sessions Monitoring',

@enabled=1,

@notify\_level\_eventlog=0,

@notify\_level\_email=2,

@notify\_level\_netsend=0,

@notify\_level\_page=0,

@delete\_level=0,

@description=N'Inserts log records when number of connections is higher than 50',

@category\_name=N'DBA',

@owner\_login\_name=N'sa',

@job\_id = @jobId OUTPUT

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

EXEC @ReturnCode = msdb.dbo.sp\_add\_jobstep @job\_id=@jobId, @step\_name=N'Log info',

@step\_id=1,

@cmdexec\_success\_code=0,

@on\_success\_action=1,

@on\_success\_step\_id=0,

@on\_fail\_action=2,

@on\_fail\_step\_id=0,

@retry\_attempts=0,

@retry\_interval=0,

@os\_run\_priority=0, @subsystem=N'TSQL',

@command=N'INSERT INTO perf\_warehouse.dbo.\_demo\_sessions\_alert

SELECT host\_name, program\_name, login\_name, count(c.session\_id ) num\_sessions, getdate()

FROM sys.dm\_exec\_connections c JOIN sys.dm\_exec\_sessions s on c.session\_id = s.session\_id

GROUP BY host\_name,program\_name,login\_name ORDER BY 4 DESC',

@database\_name=N'master',

@flags=0

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

EXEC @ReturnCode = msdb.dbo.sp\_update\_job @job\_id = @jobId, @start\_step\_id = 1

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

EXEC @ReturnCode = msdb.dbo.sp\_add\_jobserver @job\_id = @jobId, @server\_name = N'(local)'

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

COMMIT TRANSACTION

GOTO EndSave

QuitWithRollback:

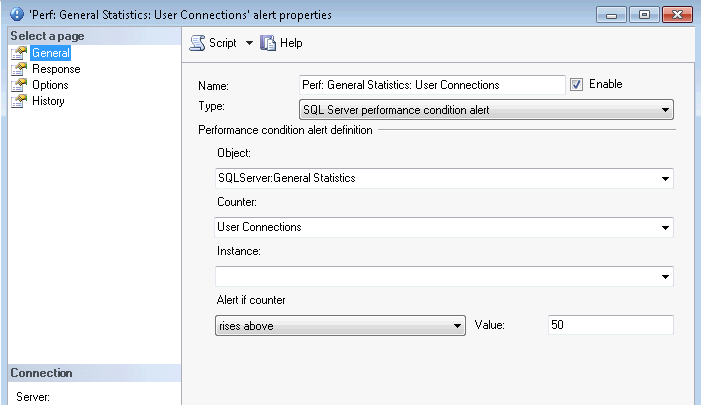
IF (@@TRANCOUNT > 0) ROLLBACK TRANSACTION

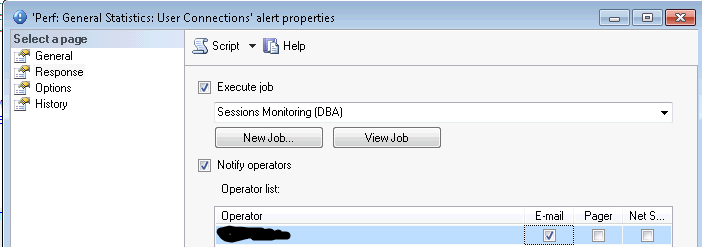
EndSave:

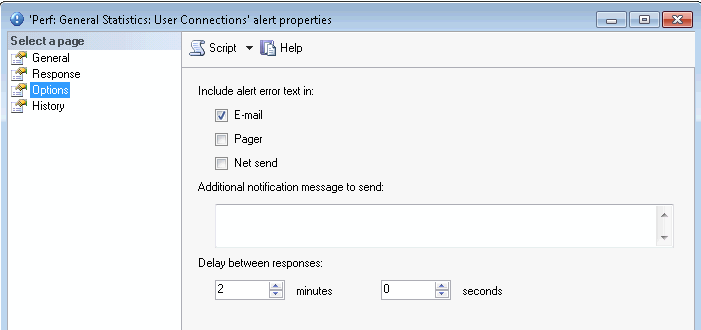
GO

**Create a SQL Server Agent alert**

Now we will create a performance condition alert that will be triggered when there are more than 50 connections (with 2 minute intervals). This alert will start the job we created above:







Here is the complete script to create the alert (you will need to update parameter "@operator\_name" with your value):

EXEC msdb.dbo.sp\_add\_alert @name=N'Perf: General Statistics: User Connections',

@message\_id=0,

@severity=0,

@enabled=1,

@delay\_between\_responses=120,

@include\_event\_description\_in=1,

@performance\_condition=N'SQLServer:General Statistics|User Connections||>|50',

@job\_name=N'Sessions Monitoring'

GO

EXEC msdb.dbo.sp\_add\_notification @alert\_name=N'Perf: General Statistics: User Connections',

@operator\_name=N'DBA\_Operator', -- update with your value

@notification\_method = 1

GO

**Testing the Alert**

Now we should be able to receive email notifications when there are more than 50 server connections and we can review the log table that contains the details:

SELECT TOP 5 [host\_name]

,[program\_name]

,[login\_name]

,[num\_sessions]

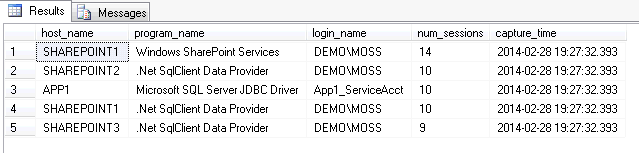
,[capture\_time]

FROM [dbo].[\_demo\_sessions\_alert]

ORDER BY [capture\_time] DESC, [num\_sessions] DESC

GO

Below is the output and we can see the application and host that had the largest number of connections:



A value of 50 connections could be low for your server and you may get a lot of alerts. For our issue to track applications not closing connections we used a value of 1000 connections.

**Next Steps**

* Use this tip to monitor your connections and to get a baseline for the number of connections your SQL Server usually has.
* After collecting some data, adjust your monitored number of connections in the alert (for example add 100 to the maximum connections in your baseline).
* Read this [tip](http://www.mssqltips.com/sqlservertip/2551/automate-sql-server-monitoring-with-email-alerts) about How to Automate SQL Server Monitoring with Email Alerts.
* Read this [tip](http://www.mssqltips.com/sqlservertip/1523/how-to-setup-sql-server-alerts-and-email-operator-notifications) about How to setup SQL Server alerts and email operator notifications.

Last Update: 3/28/2014

[**View all my tips**](http://www.mssqltips.com/sqlserverauthor/94/svetlana-golovko/)

Top of Form

Bottom of Form

[Learn More](http://www.mssqltips.com/sql-server-tip-category/54/monitoring/?utm_source=website&utm_medium=rightside&utm_content=learnmore&utm_campaign=20140221)

**Comments and Feedback:**

|  |  |
| --- | --- |
| **Friday, March 28, 2014 - 10:48:00 AM - Edward Pochinski** | [Read The Tip](http://www.mssqltips.com/sqlservertip/3193/automated-collection-of-sql-server-database-connections-for-monitoring/) |
| Very nicely done, I'm old school use to load performance data like this into a perfmon database like below then have a few stored procedures to surf the data and send alerts. This is one of the many objects loaded.: The below was containerd in a .vbs script a called from a SQL agent jop step  Set objConn = CreateObject("ADODB.Connection")  Set objRS = CreateObject("ADODB.Recordset")  objConn.Open  "PROVIDER=SQLOLEDB;DATA SOURCE=sqlsharkpc;UID=loginid;PWD=sqldba;DATABASE=perfMon "    objRS.CursorLocation = 3  objRS.Open "SELECT \* FROM Re\_Compiles" , objConn, 3, 3  strComputer = "."  Set objWMIService = GetObject("winmgmts:" \_      & "{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")  Set colRetrievedEvents = objWMIService.ExecQuery \_      ("Select \* from Win32\_PerfRawData\_MSSQLSERVER\_SQLServerSQLStatistics",,48)  For Each objEvent in colRetrievedEvents      objRS.AddNew      objRS("Compiles") = objEvent.SQLReCompilationsPersec      objRS.Update  Next  objRS.Close  objConn.Close | |