**Script to check SQL Server connection pooling**

|  |  |  |
| --- | --- | --- |
| http://www.mssqltips.com/images/SankarReddy.JPG | By: [Sankar Reddy](http://www.mssqltips.com/sqlserverauthor/41/sankar-reddy/) | [Read Comments (4)](http://www.mssqltips.com/sqlservertip/1830/script-to-check-sql-server-connection-pooling/#comments) | [Print](http://www.mssqltips.com/tipprint.asp?tip=1830)   Sankar is a Database Engineer/DBA and SQL Server MVP. He has been working with SQL Server since 2003 in a variety of roles.  Related Tips: [More](http://www.mssqltips.com/sql_server_developer_tips.asp) | [http://www.mssqltips.com/images/MSSQLTipsTurnsSix_Giveaway.gif](http://www.mssqltips.com/giveaway.asp) |

**Problem**  
Our applications are using a high number of connections and the server is very busy. How do I check whether the application is using [connection pooling](http://msdn.microsoft.com/en-us/library/bb399543.aspx) correctly when connecting to SQL Server? Is there an easy way to see which connections are using connection pooling and which ones are not? In this tip, I will go over how to capture this information for each logon, so you can see which connections are taking advantage of connection polling.

**Solution**  
SQL Server 2005 introduced DDL Triggers and this solution takes advantage of the DDL Logon trigger along with the EVENTDATA() function to determine if the connections are using connection pool or NOT.

Identifying this fact is very important to every DBA as the cost of opening a connection every time data is requested is expensive. The cost of each user connection is approximately (3 \* network\_packet\_size + 94 KB) where the default network packet size is 4 KB. That means each user connection cost is approximately 130 KB. In addition, the time it takes to make the connection can also slow down processing.

Here is the script to create a table to store the data and a DDL Logon trigger to capture this data. This should be created in the master database.

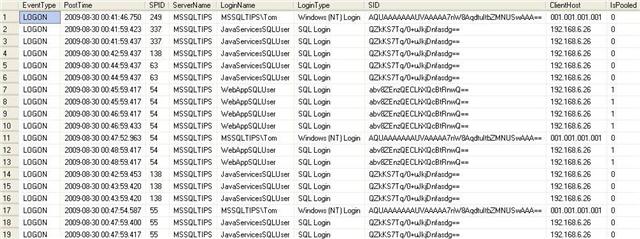
|  |
| --- |
| --Create the dbo.ServerLogonHistory Table  CREATE TABLE dbo.ServerLogonHistory  (  EventType VARCHAR(512),  PostTime DATETIME,  SPID INT,  ServerName VARCHAR(512),  LoginName VARCHAR(512),  LoginType VARCHAR(512),  SID VARCHAR(512),  ClientHost VARCHAR(512),  IsPooled BIT  )  GO  --Create the Logon Trigger Trigger\_ServerLogon  CREATE TRIGGER Trigger\_ServerLogon  ON ALL SERVER WITH EXECUTE AS 'sa' FOR LOGON  AS  BEGIN  DECLARE @data XML  SET @data = EVENTDATA()  INSERT INTO dbo.ServerLogonHistory  SELECT @data.value('(/EVENT\_INSTANCE/EventType)[1]', 'nvarchar(512)')  , @data.value('(/EVENT\_INSTANCE/PostTime)[1]', 'datetime')  , @data.value('(/EVENT\_INSTANCE/SPID)[1]', 'nvarchar(4)')  , @data.value('(/EVENT\_INSTANCE/ServerName)[1]', 'nvarchar(512)')  , @data.value('(/EVENT\_INSTANCE/LoginName)[1]', 'nvarchar(512)')  , @data.value('(/EVENT\_INSTANCE/LoginType)[1]', 'nvarchar(512)')  , @data.value('(/EVENT\_INSTANCE/SID)[1]', 'nvarchar(512)')  , @data.value('(/EVENT\_INSTANCE/ClientHost)[1]', 'nvarchar(512)')  , @data.value('(/EVENT\_INSTANCE/IsPooled)[1]', 'nvarchar(1)')  END  GO |

In order for users to write to this table, that do not have rights to this table, I am using the [EXECUTE AS](http://www.mssqltips.com/sqlservertip/1227/granting-permission-with-the-execute-as-command-in-sql-server/) clause in the trigger to run the code using the equivalent of the "sa" account. If you do not want to use the "sa" account another option is to create a new login, grant insert permissions to this table for this login and then use this login in the EXECUTE AS clause in the trigger.

Here is another way to do this by giving [GRANT](http://www.mssqltips.com/sqlservertip/1138/giving-and-removing-permissions-in-sql-server/) INSERT permissions to PUBLIC for table master.dbo.ServerLogonHistory.

|  |
| --- |
| --Create the dbo.ServerLogonHistory Table  CREATE TABLE dbo.ServerLogonHistory  (  EventType VARCHAR(512),  PostTime DATETIME,  SPID INT,  ServerName VARCHAR(512),  LoginName VARCHAR(512),  LoginType VARCHAR(512),  SID VARCHAR(512),  ClientHost VARCHAR(512),  IsPooled BIT  )  GO    --Grant insert rights to public for this table  GRANT INSERT ON dbo.ServerLogonHistory TO PUBLIC    --Create the Logon Trigger Trigger\_ServerLogon  CREATE TRIGGER Trigger\_ServerLogon  ON ALL SERVER FOR LOGON  AS  BEGIN  DECLARE @data XML  SET @data = EVENTDATA()  INSERT INTO dbo.ServerLogonHistory  SELECT @data.value('(/EVENT\_INSTANCE/EventType)[1]', 'nvarchar(512)')  , @data.value('(/EVENT\_INSTANCE/PostTime)[1]', 'datetime')  , @data.value('(/EVENT\_INSTANCE/SPID)[1]', 'nvarchar(4)')  , @data.value('(/EVENT\_INSTANCE/ServerName)[1]', 'nvarchar(512)')  , @data.value('(/EVENT\_INSTANCE/LoginName)[1]', 'nvarchar(512)')  , @data.value('(/EVENT\_INSTANCE/LoginType)[1]', 'nvarchar(512)')  , @data.value('(/EVENT\_INSTANCE/SID)[1]', 'nvarchar(512)')  , @data.value('(/EVENT\_INSTANCE/ClientHost)[1]', 'nvarchar(512)')  , @data.value('(/EVENT\_INSTANCE/IsPooled)[1]', 'nvarchar(1)')  END  GO |

Here is a sample result from the data that was collected using the above DDL trigger.

****

You can look at the IsPooled column to check if the connections are pooled.

* 1 = pooled and
* 0 = non-pooled connections.

**Next Steps:**

* Here is [another approach](http://blogs.conchango.com/jamesrowlandjones/archive/2009/07/25/how-to-tell-if-connections-to-sql-server-are-pooled-or-not.aspx) for finding connection pool information by James Rowland-Jones
* Take a look at these other [DDL Triggers](http://www.mssqltips.com/category.asp?catid=67)
* Keep on eye on this audit table, because this will track all connections to your database server and may get quite large.
* If you have any issues with LOGON triggers and you can not connect to SQL Server [review this tip](http://www.mssqltips.com/sqlservertip/1631/connecting-to-sql-server-with-a-bad-logon-trigger/)

Related Tips: [More](http://www.mssqltips.com/sql_server_developer_tips.asp) | [Become a paid author](http://www.mssqltips.com/contribute.asp)

Last Update: 9/4/2009