Whenever we write any procedure and execute it a message appears in message window that shows number of rows affected with the statement written in the procedure.

But this message creates an extra overhead on the network. By using SET NOCOUNT we can remove this extra overhead from the network, that can actually improve the performance of our database and our application.

When SET NOCOUNT is ON, the count is not returned. When SET NOCOUNT is OFF, the count is returned.

Example:

USE STUDENTDB

GO

SELECT \* FROM [STUDENTDB].[STUDENT];

Messege:

(664 row(s) affected)

USE STUDENTDB

GO

SET NOCOUNT ON

SELECT \* FROM [STUDENTDB].[STUDENT];

SET NOCOUNT OFF

Messege:

Command(s) completed successfully.

SET NOCOUNT NO statement can be useful in store procedures. SET NOCOUNT ON statement into store procedures can reduce network traffic, because client will not receive the message indicating the number of rows affected by T-SQL statement. Setting SET NOCOUNT to ON can provide a significant performance boost, because network traffic is greatly reduced.

The @@ROWCOUNT function is updated even when SET NOCOUNT is ON.

============

Sometimes even the simplest things can make a difference. One of these simple items that should be part of every stored procedure is [SET NOCOUNT ON](http://msdn2.microsoft.com/en-us/library/ms189837.aspx). This one line of code, put at the top of a stored procedure turns off the messages that SQL Server sends back to the client after each T-SQL statement is executed. This is performed for all SELECT, INSERT, UPDATE, and DELETE statements. Having this information is handy when you run a T-SQL statement in a query window, but when stored procedures are run there is no need for this information to be passed back to the client.

By removing this extra overhead from the network it can greatly improve overall performance for your database and application.

If you still need to get the number of rows affected by the T-SQL statement that is executing you can still use the[@@ROWCOUNT](http://msdn2.microsoft.com/en-us/library/ms187316.aspx) option. By issuing a [SET NOCOUNT ON](http://msdn2.microsoft.com/en-us/library/ms189837.aspx) this function ([@@ROWCOUNT](http://msdn2.microsoft.com/en-us/library/ms187316.aspx)) still works and can still be used in your stored procedures to identify how many rows were affected by the statement.