# C:\Users\stefano.maggi.CONBIPELSPA\Desktop\prom38.png

**Date**: 06/12/2012

**Procedure:** Capture And Display Execution Time Of SQL Query In SQL Server

**Source:** [**LINK**](http://4rapiddev.com/sql-server/capture-and-display-execution-time-of-sql-query-in-sql-server/)

**Permalink:** [**LINK**](http://heelpbook.altervista.org/2012/capture-and-display-execution-time-of-sql-query-in-sql-server/comment-page-1/)

**Created by:** HeelpBook Staff

**Document Version:** 1.0

# [**Capture And Display Execution Time Of SQL Query In SQL Server**](http://heelpbook.altervista.org/2012/capture-and-display-execution-time-of-sql-query-in-sql-server/)

This tutorial proposes 3 ways in order for you to get the **Execution time** of SQL Query or Stored Procedures are called or submitted to your SQL Server.

They will give you **durations in microseconds** and base on the execution time, you may have a deeper understand and will do some optimization for your database structure/indexing to make it runs better.

### 1. Using SQL Server Profiler

I think it’s a easiest way for you to trace/track which Stored Procedures or SQL commands are running on SQL Server and how long it takes for each of  **SQL Query/ Stored Procedure** execution.



As you see, all commands are in **TextData**column and all Execution time for each are in **Duration**column respectively.

### 2. Using SQL Script with @StartTime and @EndTime parameters

The script should be run on **SQL Server Management Studio Query**.

|  |
| --- |
| USE AdventureWorksDW;GODECLARE @StartTime datetime,@EndTime datetimeSELECT @StartTime=GETDATE()SELECT \* FROM DimCustomer where Gender = ‘M’SELECT @EndTime=GETDATE()SELECT DATEDIFF(ms,@StartTime,@EndTime) AS [Duration in microseconds] |

Just replace your own SQL statements with line 2, after execute the statement, it will show the **Duration in microseconds**in another result panel.



### 3. Using SQL Script with SET STATISTICS TIME (Transact-SQL)

It displays the number of milliseconds required to parse, compile, and execute each statement.

Run this **SQL** script on your **SQL Query**:

|  |
| --- |
| USE AdventureWorksDW;GOSET STATISTICS TIME ONGOSELECT \* FROM DimCustomer where Gender = ‘M’GoSET STATISTICS TIME OFF;GO |

And below is the result set:

|  |
| --- |
| SQL Server parse and compile time:CPU time = 0 ms, elapsed time = 1 ms.SQL Server parse and compile time:CPU time = 0 ms, elapsed time = 1 ms.(9351 row(s) affected)SQL Server Execution Times:CPU time = 63 ms, elapsed time = 479 ms.SQL Server parse and compile time:CPU time = 0 ms, elapsed time = 1 ms. |



That’s all! Feel free to contribute your own solution by submitting your comments as you are always be welcome.