In this video we will discuss **how to set the execution order of triggers** using **sp\_settriggerorder**stored procedure.   
  
   
  
**Server scoped triggers will always fire before any of the database scoped triggers**. This execution order cannot be changed.   
  
In the example below, we have a database-scoped and a server-scoped trigger handling the same event (CREATE\_TABLE). When you create a table, notice that server-scoped trigger is always fired before the database-scoped trigger.  

CREATE TRIGGER tr\_DatabaseScopeTrigger

ON DATABASE

FOR CREATE\_TABLE

AS

BEGIN

    Print 'Database Scope Trigger'

END

GO

CREATE TRIGGER tr\_ServerScopeTrigger

ON ALL SERVER

FOR CREATE\_TABLE

AS

BEGIN

    Print 'Server Scope Trigger'

END

GO

Using the **sp\_settriggerorder**stored procedure, you can set the execution order of server-scoped or database-scoped triggers.   
  
**sp\_settriggerorder stored procedure has 4 parameters**

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| @triggername | Name of the trigger |
| @order | Value can be First, Last or None. When set to None, trigger is fired in random order |
| @stmttype | SQL statement that fires the trigger. Can be INSERT, UPDATE, DELETE or any DDL event |
| @namespace | Scope of the trigger. Value can be DATABASE, SERVER, or NULL |

EXEC sp\_settriggerorder

@triggername = 'tr\_DatabaseScopeTrigger1',

@order = 'none',

@stmttype = 'CREATE\_TABLE',

@namespace = 'DATABASE'

GO

**If you have a database-scoped and a server-scoped trigger handling the same event**, and if you have set the execution order at both the levels. Here is the execution order of the triggers.  
1. The server-scope trigger marked First  
2. Other server-scope triggers  
3. The server-scope trigger marked Last  
4. The database-scope trigger marked First  
5. Other database-scope triggers  
6. The database-scope trigger marked Last