Article: Recursive SP on all Databases
Date: 14/03/2012
Posted by: HeelpBook Staff
Source: Link
Permalink: Link

SQL Server – Create Same Stored Procedure on All Databases using sp_MSForEachDB T-SQL

If you have a stored procedure which you use for database *maintenance* or database tracking you may want to create this **T-SQL** stored procedure on every database in the **MS SQL Server** instance.
Since you want to make the deployment of a **SQL Stored Procedure** easier, and you want to create stored procedure on all databases in one click, you will have to use a tool for **SQL Server** which will deploy the sp to each database for you.

Or you will develop a t-sql script which will create the stored procedure for you on all sql databases. I'll try to explain an *sql example* script which I'm using to make such deployments in my development and production database environment.

Let's build a sample stored procedure which we want to deploy on every sql database installed in the **MS SQL Server**.

```
CREATE PROC TestSpForAllDBs AS SELECT 1
```

Create this t-sql stored procedure in **master** database. Now we will create a deployment stored procedure which will get the text of this database object **TestSpForAllDBs**, and will execute this text on every database on the sql instance.

Please note that the below stored procedure named "*CreateProcedure*" will be created in **master** database of the **SQL Server**.

The stored procedure creator procedure :) takes two parameters.

- First parameter **@dbname** is the database name parameter. We will use database name parameter as an identifier showing the target database of this deployment.
- The second parameter **@spname** is the name of the stored procedure which we want to deploy on every sql database. For our sql example this sp name will be "**TestSpForAllDBs**".

```
CREATE PROC CreateProcedure
(
  @dbname sysname,
  @spname sysname
)
AS
SELECT @dbname = REPLACE(REPLACE(@dbname,'[',''),']','')
IF @dbname <> 'master'
BEGIN
DECLARE @proc_text nvarchar(max)
SELECT
  @proc_text = REPLACE([text],'''','''''')
```

```
FROM [sysobjects] o
INNER JOIN [syscomments] c
  ON c.id = o.id
WHERE
  o.type = 'P' AND
  o.name = @spname
DECLARE @sql nvarchar(max)
SET @sql = 'USE ' + @dbname + '; EXEC ('' ' + @proc_text + ''');'
EXEC sp_Executesql @sql
END
GO
```

If you take a look at the above t-sql source code of the CreateProcedure, you will notice that :

- We are reading the text or create code of the stored procedure into a **nvarchar(max)** parameter;
- Then we create a dynamic **T-SQL** command code.

This dynamic **T-SQL** code has the following properties:

- First it executes a **USE *databasename*;** syntax in order to change the executing database.
- Later, it runs the create procedure command text by using the **EXEC('*sqlcommand*')** syntax.

The above part of the solution only creates the stored procedure on a given target database. We have to manually specify the database name.

The solution of this problem is actually very easy by using the **sp_MSForEachDB** undocumented stored procedure.

All we have to do for a complete solution is as simple as calling the below **sp_MSForEachDB** command.

```
EXEC sp_MSForEachDB 'CreateProcedure ''[?]'', ''TestSpForAllDBs'''
```

As you see, the above t-sql **sp_MSForEachDB** statement will execute the *CreateProcedure* stored procedure on master database for each database in the **MS SQL Server** instance. Each execution will have a different database name value for the **@dbname** parameter.

And this difference will enable us deploy our example sql stored procedure on every database on the **SQL Server**.