Date: 13/12/2012
Procedure: Marking a stored procedure as System Object (SQL Server)
Source: LINK
Permalink: LINK
Created by: HeelpBook Staff
Document Version: 1.0

MARKING A STORED PROCEDURE AS SYSTEM OBJECT (SQL SERVER)

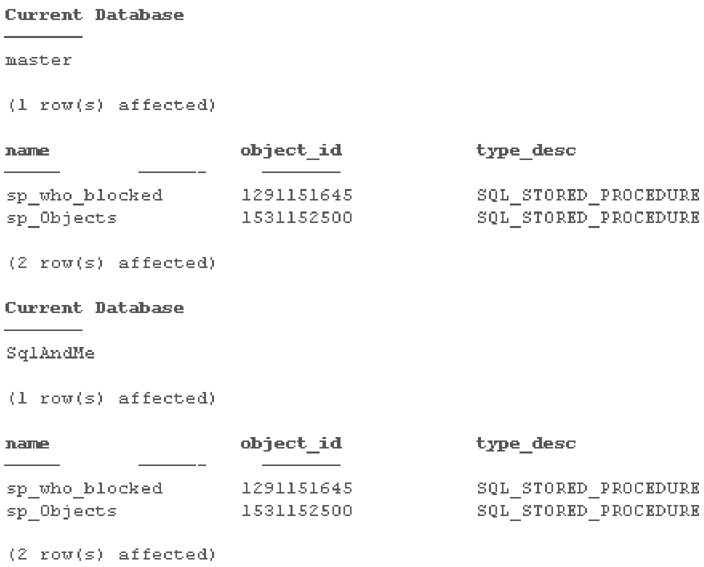
Marking a stored procedure as system object allows to run the procedure in a user database context. There are two requirements to allow running a procedure created in **[master]** database in user database context:

THE STORED PROCEDURE NAME MUST BEGIN WITH "SP_" :

A stored procedure created with **"sp_"** prefix can be used in any user database without specifying database/schema. But, the procedure still run in the context of master database and not the user database. Let's create a procedure to test this:

```
USE   [master]

GO

CREATE PROCEDURE sp_Objects

AS

SELECT      name, object_id, type_desc

FROM        sys.objects

WHERE       is_ms_shipped <> 1

GO

-Execute procedure in [master]

SELECT DB_NAME() 'Current Database'

EXEC sp_Objects

--Execute procedure in [SqlAndMe]

USE   [SqlAndMe]

SELECT DB_NAME() 'Current Database'

EXEC sp_Objects
```

RESULT SET:



```
Current Database
----------------
master

(1 row(s) affected)

name                 object_id          type_desc
----                 ---------          ---------
sp_who_blocked       1291151645         SQL_STORED_PROCEDURE
sp_Objects           1531152500         SQL_STORED_PROCEDURE

(2 row(s) affected)

Current Database
----------------
SqlAndMe

(1 row(s) affected)

name                 object_id          type_desc
----                 ---------          ---------
sp_who_blocked       1291151645         SQL_STORED_PROCEDURE
sp_Objects           1531152500         SQL_STORED_PROCEDURE

(2 row(s) affected)
```

As you can see from the result set, the procedure **sp_Objects** runs under **[master]** even after switching the database using **"USE DB"**.

THE STORED PROCEDURE MUST BE MARKED AS SYSTEM OBJECT EXPLICITLY:

You can mark a stored procedure as system object using **sys.sp_MS_marksystemobject** system procedure. Let's mark our procedure **sp_Objects** as system object and re-execute above code.

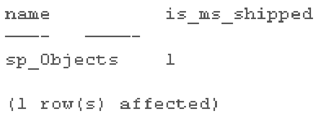
Below code will mark the procedure as system object:

```
USE [master]

EXEC sys.sp_MS_marksystemobject sp_Objects
```

You can verify if the object is marked as system object:

```
USE [master]


SELECT name, is_ms_shipped

FROM sys.objects

WHERE name = ‘sp_objects’
```

RESULT SET:

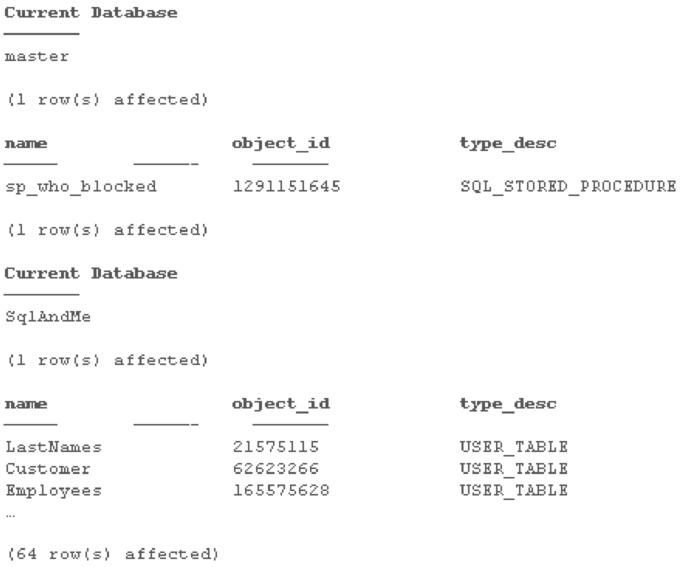


```
name          is_ms_shipped
----   ------
sp_Objects    1

(1 row(s) affected)
```

sp_Objects is now marked as system object and can be run in user database context:

```
-Execute procedure in [master]

USE [master]

SELECT DB_NAME() ‘Current Database’

EXEC sp_Objects

-Execute procedure in [SqlAndMe]
USE [SqlAndMe]

SELECT DB_NAME() ‘Current Database’

EXEC sp_Objects
```

RESULT SET:



```
Current Database
----------------
master

(1 row(s) affected)

name            ------         object_id          type_desc
----                           ---------
sp_who_blocked                 1291151645         SQL_STORED_PROCEDURE

(1 row(s) affected)

Current Database
----------------
SqlAndMe

(1 row(s) affected)

name            ------         object_id          type_desc
----                           ---------
LastNames                      21575115           USER_TABLE
Customer                       62623266           USER_TABLE
Employees                      165575628          USER_TABLE
…

(64 row(s) affected)
```

You can also create tables in master database which begin with prefix "**sp_**", and these can be used in user databases without database/schema prefix. It does not need to marked as system object. Try below example yourself:

```
–Create Table in [master]

USE [master]

GO

SELECT DB_NAME() 'Current Database'

CREATE TABLE sp_Table1

(

Col1 CHAR(10)

)

INSERT INTO sp_Table1

VALUES ('Master')

–Insert/Select from [SqlAndMe]

USE [SqlAndMe]

SELECT DB_NAME() 'Current Database'

INSERT INTO sp_Table1

VALUES ('SqlAndMe')


SELECT *

FROM sp_Table1
```

Hope This Helps!