

MSSQL Server Change Object Owner

Published 2/22/2009 by [Tim Eisenhauer](#)

I was doing a database migration from a shared hosting plan to a MOSSO database and ran into an issue regarding MSSQL object ownership.

When the database was originally installed, it was installed under a user other than the "dbo" user.

Since MOSSO has specific naming conventions, it was impossible for me to just recreate this user in their system. So, this left me with a ton of tables, stored procedures, views, and functions that were all 'owned' by a user other than the DBO – and a big headache.

Here is an example of what some of these object looked like:

[mylittleuser].[Proc_UpdateUser]

[mylittleuser].[Proc_InsertUser]

[mylittleuser].[Tbl_User]

Now, there were over 50 tables, a ton of functions, views, and over 100 stored procedures that were not 'owned' by the dbo user. So I wasn't about to go to each object and update the schema. I scoured the internet looking for a script to do this and had very little luck.

There were a lot of scripts that would update only the tables...or only the sprocs...or only the views. I couldn't find anything that would update everything, including PK's, FK's tables, views, sprocs, functions, etc.

When I was just about to bang my head against the wall and hope the script would fall out and paste itself into the query window, I came upon a page on Microsoft's website. I was saved and born again! Woohoo! It work great.

<http://support.microsoft.com/kb/275312>

```
if exists (select * from sysobjects where id = object_id(N'[dbo].[chObjOwner]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[chObjOwner]
GO

SET QUOTED_IDENTIFIER OFF      SET ANSI_NULLS ON
GO

CREATE proc chObjOwner( @usrName varchar(20), @newUserName varchar(50))
as
-- @usrName is the current user
-- @newUserName is the new user

set nocount on
declare @uid int                -- UID of the user
declare @objName varchar(50)    -- Object name owned by user
declare @currObjName varchar(50) -- Checks for existing object owned by new user
declare @outStr varchar(256)    -- SQL command with 'sp_changeobjectowner'
set @uid = user_id(@usrName)

declare chObjOwnerCur cursor static
for
select name from sysobjects where uid = @uid

open chObjOwnerCur
if @@cursor_rows = 0
begin
    print 'Error: No objects owned by ' + @usrName
    close chObjOwnerCur
    deallocate chObjOwnerCur
    return 1
end

fetch next from chObjOwnerCur into @objName

while @@fetch_status = 0
begin
    set @currObjName = @newUserName + "." + @objName
    if (object_id(@currObjName) > 0)
        print 'WARNING *** ' + @currObjName + ' already exists ***'
```

```

    set @outStr = "sp_changeobjectowner '" + @usrName + "." + @objName + ',' + @newUserName + '"
    print @outStr
    print 'go'
    fetch next from chObjOwnerCur into @objName
end

close chObjOwnerCur
deallocate chObjOwnerCur
set nocount off
return 0

GO
SET QUOTED_IDENTIFIER OFF    SET ANSI_NULLS ON
GO

```

So, basically, what you do is this.... grab the SQL stored procedure they provide for you. Paste it into the query window and execute it. It will install a stored procedure. Once this stored procedure is installed, you'll want to execute the procedure as so:

```
exec chObjOwner 'test','dbo'
```

The 'test' above should be the OLD username that owns the objects, and the 'dbo' is the new one. So, copy/paste this into a new query window, update the usernames, and execute. Now, make sure that you choose to show the results as TEXT.

When you get the results from this stored procedure, it will provide a long list of SQL that you can then paste into yet another query window.

These are the commands that will update all of your objects. You'll have to put them into another query window and execute them to update the schema.

Now, I was still left with some issues. Some of the sprocs did not update correctly so yes, I was forced to go in and hand edit some of them. Luckily there were only 3 of them.

Now I'm onto the testing...wish me luck.

----- UPDATE -----

I thought that I'd add the code for you to change the schema of just the tables

```
exec sp_MSforeachtable "ALTER SCHEMA dbo TRANSFER ? PRINT '? modified' "
```

You'll have to change the 'dbo' to the new schema you would like the tables to use.