

SQL Server DB Refresh from Source to Destination - Article

Note:

For SQL Server 2008 to SQL Server 2014/2016, Post DB Refresh, Please change DB Compatibility level to latest version.

If you like my knowledge sharing, please do a small contribution of your choice by clicking on below links

<https://www.buymeacoffee.com/PMSQLDBA>

<https://ko-fi.com/praveenmadupu>

@ Production SQL Server Instance

@ Production Database:

STEP-1: Perform the production database full backup with COPY_ONLY.

--TSQL SCRIPT:

```
BACKUP DATABASE ProdDB
TO DISK = 'E:\Source\MSSQL\BACKUPS\ProdDB_FULL_BKP_DEC262015_12.40PM.BAK'
WITH COPY_ONLY
```

STEP-2: Move this backup file to Development Server using the below command.

Go to RUN.

Type the command ---- `\\DestinationServerName\E$\Destination\MSSQL\BACKUPS`

@ Development SQL Server Instance

--Perform full backup if required as per the application team confirmation
--Also check whether we have enough space in Backup Drive (example: E:\MSSQL\Backups)

STEP-3: Perform the development database full backup normally.

--No need of copy_only option for development databases.

```
BACKUP DATABASE DevDB
TO DISK = 'E:\Destination\MSSQL\BACKUPS\DevDB_FULL_BKP_DEC262015_12.50PM.BAK'
```

STEP-4: Run the below script on Development to extract DB Users, Roles, Object Level Permissions.

--Extracting DB Users before db refresh at test/dev/stage.

--http://www.coresql.in/how_to_script_object_level_permission.html

--Extracting DB Users from the required Development Database.

```
SET nocount ON

SELECT scripts AS '--Scripts'
FROM (SELECT Getdate() AS ScriptDateTime,
      'CREATE USER [' + DP.name + '] FOR LOGIN ['
      + SP.name + ']' + CASE WHEN DP.type_desc != 'WINDOWS_GROUP' THEN
      ' WITH DEFAULT SCHEMA = [' + Isnull(DP.default_schema_name, 'dbo') + ']'
      --+ CHAR(13)+CHAR(10)+'GO'
      ELSE '--+ CHAR(13)+CHAR(10)+'GO'
      END
      AS Scripts
FROM sys.database_principals DP,
sys.server_principals SP
WHERE SP.sid = DP.sid
AND DP.name NOT IN ( 'DBO', 'GUEST', 'INFORMATION_SCHEMA', 'SYS',
                    'PUBLIC', 'DB_OWNER', 'DB_ACCESSADMIN',
                    'DB_SECURITYADMIN',
                    'DB_DDLADMIN', 'DB_BACKUPOPERATOR', 'DB_DATAREADER'
                    ,
                    'DB_DATAWRITER',
                    'DB_DENYDATAREADER', 'DB_DENYDATAWRITER', 'DB_X' )

UNION
```

--Extracting Database Roles Permissions for the DB Users.

```
SELECT Getdate() AS ScriptDateTime,
'EXEC sp_addrolemember @rolename ='
+ Space(1)
+ Quotename(User_name(rm.role_principal_id), '')
+ ', @membername =' + Space(1)
+ Quotename(User_name(rm.member_principal_id), '')
--+ CHAR(13)+CHAR(10)+'GO'
AS '--Role Memberships'
FROM sys.database_role_members AS rm
WHERE User_name(rm.role_principal_id)
+ User_name(rm.member_principal_id) != 'DB_OWNERDBO'
--ORDER BY rm.role_principal_id ASC
UNION
```

--Extracting object level permissions

```
SELECT Getdate() AS ScriptDateTime,
CASE WHEN perm.state <> 'W' THEN perm.state_desc ELSE 'GRANT' END +
Space
(1) +
perm.permission_name + Space(1)
+ 'ON ' + Quotename(User_name(obj.schema_id))
+ '.' + Quotename(obj.name) + CASE WHEN cl.column_id IS NULL THEN Space(
0
) ELSE
'(' + Quotename(cl.name) + ')' END + Space(1) + 'TO'
+ Space(1)
+ Quotename(User_name(usr.principal_id)) COLLATE database_default + CASE
WHEN perm.state <> 'W' THEN Space(0)
ELSE Space(1) + 'WITH GRANT OPTION'

END
--+ CHAR(13)+CHAR(10)+'GO'
AS '--Object Level Permissions'
FROM sys.database_permissions AS perm
INNER JOIN sys.objects AS obj
ON perm.major_id = obj.[object_id]
INNER JOIN sys.database_principals AS usr
ON perm.grantee_principal_id = usr.principal_id
LEFT JOIN sys.columns AS cl
ON cl.column_id = perm.minor_id
AND cl.[object_id] = perm.major_id
--ORDER BY perm.permission_name ASC, perm.state_desc ASC
UNION
```

--Extracting database level permissions

```

SELECT Getdate() AS ScriptDateTime,
CASE WHEN perm.state <> 'W' THEN perm.state_desc ELSE 'GRANT' END +
Space
(1) +
perm.permission_name + Space(1)
+ Space(1) + 'TO' + Space(1)
+ Quotename(User_name(usr.principal_id)) COLLATE database_default + CASE
WHEN perm.state <> 'W' THEN Space(0)
ELSE Space(1) + 'WITH GRANT OPTION'
END
--+ CHAR(13)+CHAR(10)+'GO'
AS '--Database Level Permissions'
FROM sys.database_permissions AS perm
INNER JOIN sys.database_principals AS usr
ON perm.grantee_principal_id = usr.principal_id
WHERE perm.major_id = 0
AND ( permission_name
+ User_name(usr.principal_id) != 'CONNECTDBO' )
--ORDER BY perm.permission_name ASC, perm.state_desc ASC
) AS UserScripts
ORDER BY scripts

```

Save the OUTPUT of the above script on the Development Server as 'D:\MSSQL\DevDB_Objects.txt'

STEP-5: Now restore the Production Database Backup file on top of Development DB as follows.

--Restore with replace.

```

RESTORE DATABASE DevDB
FROM DISK = 'E:\Destination\MSSQL\BACKUPS\ProdDB_FULL_BKP_DEC262015_12.40PM.BAK'
WITH REPLACE,
Move 'DevDB' to 'E:\Destination\MSSQL\Data\DevDB_Data.mdf',
Move 'DevDB_Log' to 'O:\Destination\MSSQL\Log\DevDB_Log.ldf'

```

--Monitor the DB Restore Remaining Time Report:

--<http://www.sanssql.com/2008/11/query-to-find-time-remaining-to.html>

```

USE master
GO

```

```

SELECT
    Percent_Complete,
    Start_Time ,
    Command,
    b.Name AS DatabaseName, --Sometimes this will be "Main" as the database will not be accesiable.
    DATEADD(ms,estimated_completion_time,GETDATE()) AS RemainTime,
    (estimated_completion_time/1000/60) AS MinutesToFinish
FROM sys.dm_exec_requests a
    INNER JOIN sys.databases b
    ON a.database_id = b.database_id
WHERE
    Command like '%Restore%'

```

OR Command like '%Backup%'
AND Estimated_Completion_Time > 0

@ Development Server:

STEP-6: Delete the production db users **which are now available in Development database as orphan users.**

--Cross check for any orphan users using below script

```
USE DevDB  
GO  
SP_CHANGE_USERS_LOGIN 'REPORT'
```

STEP-7: Use the Script **DevDB_Objects.txt** taken on Dev DB and execute it on Dev DB "DevDB".

DevDB_Objects Script Path: D:\MSSQL\DevDB_Objects.txt

Note: Again check for any orphan users, if any exists fix them.

STEP-8: Finally cross the DB Roles/permissions for the respective DB users/Logins on Development DB -- DevDB.

If you like my knowledge sharing, please do a small contribution of your choice by clicking on below links

<https://www.buymeacoffee.com/PMSQLDBA>

<https://ko-fi.com/praveenmadupu>