Find Replaces Logic

**Objective**: Below logic is useful for e writing find replace logic on objects of type Procedure, Trigger and Views.

**Problem Statement**: In case of SQL 2008 to SQL 2016 migration assuming the remediation includes replacement of a syntax with another. Below SQL code will help us to find a syntax in Object like (Procedure, Trigger and Views) and generate the updated script.

**Functional Approach:** A Master table is created to accept list of find and replace string syntax , below script will look up Definition of Procedures , Views and Trigger for find string and replace it with the replace string and generate a script as output .

**Technical Approach:**

1. Create a Master Table TBL\_REMIDIATION\_FNRMASTER Developer can add the find and replace string in master table using below syntax. Where Id is unique increment in sequence like (1,2,3..), FSTRING is string to be find and RSTRING string to be replaced.

Insert into TBL\_REMIDIATION\_FNRMASTER (ID, FSTRING, and RSTRING) values (1,’varchar (100)',’Varchar (5000)')

Note: Above example is just for reference where you can replace the FSTRING and RSTRING with actual migration syntax.

1. Execute Below script in SSMS after connecting to the target database.

--=================================================Remediation Automation ======================================================

use master

if exists ( select \* from sys.types where name = 'test')

begin

drop type test

end

create type test as table (id int ,ObjectId bigint ,val nvarchar(max))

declare @CCount bigint

declare @tbl test

declare @Count int

insert into @tbl

select row\_number()over(order by object\_id ) id ,

object\_id, definition

from sys.sql\_modules

select @Count=count(\*) from @tbl

while @Count <>0

begin

select @CCount=count(\*) from TBL\_REMIDIATION\_FNRMASTER

declare @query nvarchar(max)

select @query=val from @tbl where id=@Count

while @CCount<>0

begin

select @query=replace(@query,FSTRING,RSTRING) from TBL\_REMIDIATION\_FNRMASTER where id=@ccount

--exec sp\_executeSQL @query

set @Count=@Count-1

set @CCount=@CCount-1

end

print @query

end

1. Sample Output.

Current procedure definition Example

Alter Procedure [dbo].[sp\_CreateCommand]

@jsondirectory varchar(40),

@DatabaseName varchar(max),

@ServerName varchar(100),

@UserName varchar(100),

@Password varchar(100),

@TargetName varchar(100)

Script output after running above script

Alter Procedure [dbo].[sp\_CreateCommand]

@jsondirectory varchar(40),

@DatabaseName varchar(max),

@ServerName varchar(5000),

@UserName varchar(5000),

@Password varchar(5000),

@TargetName varchar(5000)

**Limitations:**

1. Need to manually connect the target database and run the script
2. It is been observed the script take considerable time to generate output, user can avoid it by filtering type of object and then getting the script.
3. All the database objet are considered and generated in script, even if there is no fins string found in object.
4. This approach is applicable for Procedures, views and Triggers only.

**Further Development to be done:**

1. Some SSIS package or .net code to be develop to call above script and get output
2. Script to be run for different type of object so that the entire time taken to generate output should get reduce.
3. Changes in SQL Script to make the script to consider only match objects
4. Separate approach and automation to be develop for other object changes excluding Procedure, Trigger and views.