Utility Settings, Features, and Scripts for SQL Server

Contents

[Document Purpose 3](#_Toc412106411)

[Version History 3](#_Toc412106412)

[Create Keyboard Shortcuts 4](#_Toc412106413)

[Functionality 4](#_Toc412106414)

[Example 4](#_Toc412106415)

[Configuration 4](#_Toc412106416)

[dbo.usp\_GenTempTableHelperStatements 6](#_Toc412106417)

[Functionality: 6](#_Toc412106418)

[Example 6](#_Toc412106419)

[Configuration: 7](#_Toc412106420)

# Document Purpose

This document provides documentation on SQL Server Management Studio (SSMS) features, utility scripts for producing boilerplate code and other useful pieces designed to simplify common activities when working in SSMS. It is meant as a living document, and all may contribute to it.

# Version History

|  |  |  |
| --- | --- | --- |
| Version | Date | Notes |
| 1.0 | Feb 19, 2015 | * Added tip on keyboard shortcut use * Added dbo.usp\_GenTempTableHelperStatements |
|  |  |  |

# Create Keyboard Shortcuts

## Functionality

Keyboard shortcuts give you fast access to common SQL commands.

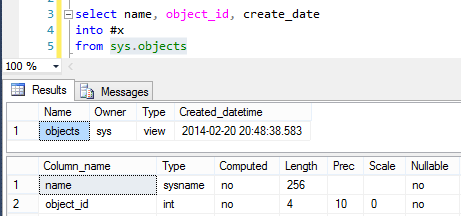
The command is prefixed to any text selected in a query window.

So, if Alt-F1 is sp\_help, then selecting sys.objects and hitting Alt-F1 executes “sp\_help sys.objects”.

This can be very useful for quick lookups when working with existing queries, since you can select the portion of text you need, without writing a separate command in another window or leaving unneeded code in your scripts.

## Example

I can select sys.objects as part of a larger query to execute sp\_help.



## Configuration

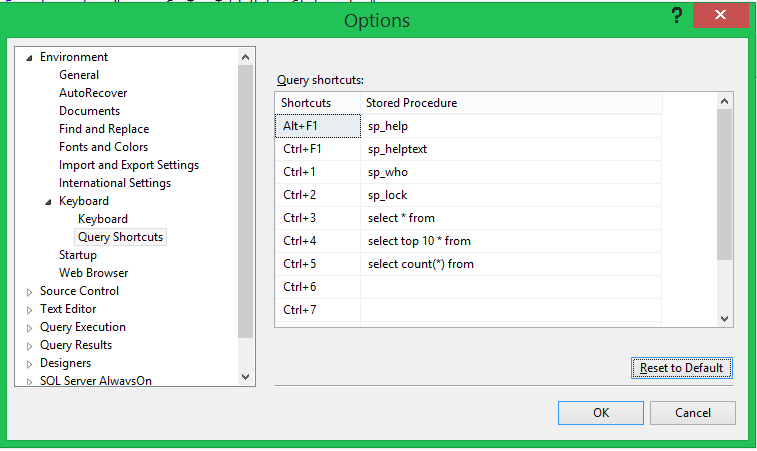
In SQL Server Management Studio, go to the Tools menu and select ‘Options’

Under Environment, select Keyboard, then Keyboard Shortcuts

Enter shortcut commands of interest. The screen shot below lists several useful examples.

Note that, as noted above, the text entered here is prefixed to your selected text. So, you will need to be sure to include a space at the end of your shortcut in most cases.

Keyboard shortcuts are available for the next new query window opened after they are created.



The “Select” options can be very useful for quick data reviews, as well as troubleshooting during query development. For example, if I were getting too few or too many rows from the following query, I can return counts (select count(\*) from ) for the first table, then include just the second table, then add the third table…, by highlighting just the portion I want.

from (select distinct \* from #table\_names\_full ) t

join sys.schemas s on s.schema\_id = T.schema\_id

left join sys.indexes i on i.object\_id = t.object\_id

left join sys.data\_spaces FG on FG.data\_space\_id = i.data\_space\_id AND FG.TYPE = 'FG' -- FILEGROUP

# dbo.usp\_GenTempTableHelperStatements

## Functionality:

This script provides several outputs that are commonly used when working with temp tables.

For an existing temp table, the script produces the following outputs:

1. If Exists… Drop
2. Create #Table (field list). Note that constraints or indices are not reproduced.
3. Select statement
4. Table variable declaration
5. Select statement returning default values by data type
6. Select statement returning null values cast to data type

## Example Output

1. Create the following temp table in any database

select name, object\_id, create\_date

into #x

from sys.objects

1. Execute master.dbo.usp\_GenTempTableHelperStatements #x

OR Set up a keyboard shortcut for

master.dbo.usp\_GenTempTableHelperStatements (+ space at end). See earlier documentation for how to create keyboard shortcuts.

The following outputs are produced.

IF OBJECT\_ID(N'TEMPDB..#x') IS NOT NULL DROP TABLE #x

-----------------------------------------------------------------------------

CREATE TABLE #x

( name nvarchar(128)

, object\_id int

, create\_date datetime

)

-----------------------------------------------------------------------------

SELECT

name

, object\_id

, create\_date

FROM #x

-----------------------------------------------------------------------------

DECLARE @x TABLE

( name nvarchar(128)

, object\_id int

, create\_date datetime

)

-----------------------------------------------------------------------------

SELECT

Cast('' as nvarchar(128)) as name

, Cast(0 as int) as object\_id

, Cast(GetDate() as datetime) as create\_date

-----------------------------------------------------------------------------

SELECT

Cast(null as nvarchar(128)) as name

, Cast(null as int) as object\_id

, Cast(null as datetime) as create\_date

## Configuration:

Run the script below to create a stored procedure in the MASTER database. (Another database can be chosen.)

Create keyboard shortcut.

## Script

USE [master]

GO

CREATE PROCEDURE [dbo].[usp\_GenTempTableHelperStatements]

(

@TABLE\_NAME VARCHAR(MAX)

)

AS begin

/\*

exec dbo.[usp\_GenTempTableHelperStatements] '#TABLE'

\*/

SET NOCOUNT ON

CREATE TABLE #TABLE

(

CREATE\_STATEMENT NVARCHAR(MAX)

,COLUMN\_ID INT

)

CREATE TABLE #SELECT\_COLUMNS

(

CREATE\_STATEMENT NVARCHAR(MAX)

,COLUMN\_ID INT

)

CREATE TABLE #DECLARE

(

CREATE\_STATEMENT NVARCHAR(MAX)

,COLUMN\_ID INT

)

CREATE TABLE #SELECT\_DEFAULT\_DATA\_TYPE

(

CREATE\_STATEMENT NVARCHAR(MAX)

, COLUMN\_ID INT

, COLUMN\_NAME NVARCHAR(500)

)

CREATE TABLE #SELECT\_NULL

(

CREATE\_STATEMENT NVARCHAR(MAX)

, COLUMN\_ID INT

, COLUMN\_NAME NVARCHAR(500)

)

-----------------------------------------------------------------------------------

DECLARE @NUM\_COLUMNS AS INT = 0, @COUNT INT, @I INT, @STMT NVARCHAR(MAX)

SELECT @NUM\_COLUMNS = max\_column\_id\_used

FROM tempdb.sys.tables

WHERE object\_id = OBJECT\_ID('tempdb.dbo.' + @TABLE\_NAME)

IF (@NUM\_COLUMNS = 0)

BEGIN

print 'Temporary table "tempdb.dbo.' + @TABLE\_NAME + '" not found.'

return

END

-----------------------------------------------------------------------------------

DECLARE @MAX\_COL\_NAME\_LENGTH INT =

(SELECT MAX(LEN(NAME))

From

tempdb.sys.columns Where object\_id=OBJECT\_ID('tempdb.dbo.' + @TABLE\_NAME))

-----------------------------------------------------------------------------------

INSERT INTO #TABLE (CREATE\_STATEMENT, COLUMN\_ID)

SELECT

'IF OBJECT\_ID(N''TEMPDB..' + @TABLE\_NAME + ''') IS NOT NULL DROP TABLE ' + @TABLE\_NAME

,1 AS column\_id

UNION ALL

SELECT '', 2

UNION ALL

SELECT

'CREATE TABLE ' + @TABLE\_NAME

,3 AS column\_id

UNION ALL

SELECT

'('

,4

UNION ALL

SELECT

CASE

WHEN Size IS NULL THEN

CASE column\_id

WHEN 1 THEN ' ' + COLUMN\_NAME + REPLICATE(' ', @MAX\_COL\_NAME\_LENGTH + 1 - LEN(COLUMN\_NAME)) + COLUMN\_TYPE

ELSE ' , ' + COLUMN\_NAME + REPLICATE(' ', @MAX\_COL\_NAME\_LENGTH + 1 - LEN(COLUMN\_NAME)) + COLUMN\_TYPE

END

ELSE

CASE column\_id

WHEN 1 THEN ' ' + COLUMN\_NAME + REPLICATE(' ', @MAX\_COL\_NAME\_LENGTH + 1 - LEN(COLUMN\_NAME)) + COLUMN\_TYPE + '(' + CAST(size AS varchar) + ')'

ELSE ' , ' + COLUMN\_NAME + REPLICATE(' ', @MAX\_COL\_NAME\_LENGTH + 1 - LEN(COLUMN\_NAME)) + COLUMN\_TYPE + '(' + CAST(size AS varchar) + ')'

END

END AS CREATE\_STATEMENT

,column\_id + 4 AS column\_id

FROM

(

SELECT

A.name AS COLUMN\_NAME

,B.name AS COLUMN\_TYPE

,A.column\_id

,CASE

WHEN B.name = 'nvarchar' THEN CASE WHEN A.max\_length = -1 THEN 'MAX' ELSE CAST(A.max\_length/2 AS VARCHAR) END

WHEN B.name = 'char' THEN CAST(A.max\_length AS VARCHAR)

WHEN B.name = 'varchar' THEN CASE WHEN A.max\_length = -1 THEN 'MAX' ELSE CAST(A.max\_length AS VARCHAR) END

WHEN B.name = 'varbinary' THEN CASE WHEN A.max\_length = -1 THEN 'MAX' ELSE CAST(A.max\_length AS VARCHAR) END

WHEN B.name = 'binary' THEN CAST(A.max\_length AS VARCHAR)

WHEN B.name = 'nchar' THEN CAST(A.max\_length/2 AS VARCHAR)

WHEN B.name = 'datetime2' THEN CAST(A.scale AS VARCHAR)

WHEN B.name = 'datetimeoffset' THEN CAST(A.scale AS VARCHAR)

WHEN B.name = 'time' THEN CAST(A.scale AS VARCHAR)

WHEN B.name = 'float' THEN CAST(A.precision AS VARCHAR)

WHEN B.name = 'decimal' THEN CAST(A.precision AS VARCHAR) + ',' + CAST(A.scale AS VARCHAR)

WHEN B.name = 'numeric' THEN CAST(A.precision AS VARCHAR) + ',' + CAST(A.scale AS VARCHAR)

ELSE NULL

END AS Size

FROM

(

Select

\*

From

tempdb.sys.columns Where object\_id=OBJECT\_ID('tempdb.dbo.' + @TABLE\_NAME)

) A

JOIN sys.types B ON A.system\_type\_id = B.system\_type\_id AND B.name <> 'sysname'

) sub\_query

UNION ALL

SELECT

')'

,@NUM\_COLUMNS + 5

ORDER BY

column\_id

SELECT @COUNT = Count(\*) FROM #TABLE

SET @I = 1

WHILE(@I <= @COUNT)

BEGIN

SELECT @STMT = CREATE\_STATEMENT FROM #TABLE WHERE COLUMN\_ID = @I

print @STMT

SET @I = @I + 1

END

PRINT ' '

PRINT '-----------------------------------------------------------------------------'

PRINT ' '

INSERT INTO #SELECT\_COLUMNS

SELECT

'SELECT'

,1 AS column\_id

UNION ALL

SELECT

CASE column\_id

WHEN 1 THEN ' ' + COLUMN\_NAME

ELSE ' , ' + COLUMN\_NAME

END AS CREATE\_STATEMENT

,column\_id + 1 AS column\_id

FROM

(

SELECT

A.name AS COLUMN\_NAME

,A.column\_id

FROM

(

Select

\*

From

tempdb.sys.columns Where object\_id=OBJECT\_ID('tempdb.dbo.' + @TABLE\_NAME)

) A

JOIN sys.types B ON A.system\_type\_id = B.system\_type\_id AND B.name <> 'sysname'

) sub\_query

UNION ALL

SELECT 'FROM ' + @TABLE\_NAME, @NUM\_COLUMNS + 2

ORDER BY

column\_id

SELECT @COUNT = Count(\*) FROM #SELECT\_COLUMNS

SET @I = 1

WHILE(@I <= @COUNT)

BEGIN

SELECT @STMT = CREATE\_STATEMENT FROM #SELECT\_COLUMNS WHERE COLUMN\_ID = @I

print @STMT

SET @I = @I + 1

END

PRINT ' '

PRINT '-----------------------------------------------------------------------------'

PRINT ' '

INSERT INTO #DECLARE

SELECT

'DECLARE @' + Substring(@TABLE\_NAME, 2, 99) + ' TABLE'

,1 AS column\_id

UNION ALL

SELECT

'('

,2

UNION ALL

SELECT

CASE

WHEN Size IS NULL THEN

CASE column\_id

WHEN 1 THEN ' ' + COLUMN\_NAME + REPLICATE(' ', @MAX\_COL\_NAME\_LENGTH + 1 - LEN(COLUMN\_NAME)) + COLUMN\_TYPE

ELSE ' , ' + COLUMN\_NAME + REPLICATE(' ', @MAX\_COL\_NAME\_LENGTH + 1 - LEN(COLUMN\_NAME)) + COLUMN\_TYPE

END

ELSE

CASE column\_id

WHEN 1 THEN ' ' + COLUMN\_NAME + REPLICATE(' ', @MAX\_COL\_NAME\_LENGTH + 1 - LEN(COLUMN\_NAME)) + COLUMN\_TYPE + '(' + CAST(size AS varchar) + ')'

ELSE ' , ' + COLUMN\_NAME + REPLICATE(' ', @MAX\_COL\_NAME\_LENGTH + 1 - LEN(COLUMN\_NAME)) + COLUMN\_TYPE + '(' + CAST(size AS varchar) + ')'

END

END AS CREATE\_STATEMENT

,column\_id + 2 AS column\_id

FROM

(

SELECT

A.name AS COLUMN\_NAME

,B.name AS COLUMN\_TYPE

,A.column\_id

,CASE

WHEN B.name = 'nvarchar' THEN CASE WHEN A.max\_length = -1 THEN 'MAX' ELSE CAST(A.max\_length/2 AS VARCHAR) END

WHEN B.name = 'char' THEN CAST(A.max\_length AS VARCHAR)

WHEN B.name = 'varchar' THEN CASE WHEN A.max\_length = -1 THEN 'MAX' ELSE CAST(A.max\_length AS VARCHAR) END

WHEN B.name = 'varbinary' THEN CASE WHEN A.max\_length = -1 THEN 'MAX' ELSE CAST(A.max\_length AS VARCHAR) END

WHEN B.name = 'binary' THEN CAST(A.max\_length AS VARCHAR)

WHEN B.name = 'nchar' THEN CAST(A.max\_length/2 AS VARCHAR)

WHEN B.name = 'datetime2' THEN CAST(A.scale AS VARCHAR)

WHEN B.name = 'datetimeoffset' THEN CAST(A.scale AS VARCHAR)

WHEN B.name = 'time' THEN CAST(A.scale AS VARCHAR)

WHEN B.name = 'float' THEN CAST(A.precision AS VARCHAR)

WHEN B.name = 'decimal' THEN CAST(A.precision AS VARCHAR) + ',' + CAST(A.scale AS VARCHAR)

WHEN B.name = 'numeric' THEN CAST(A.precision AS VARCHAR) + ',' + CAST(A.scale AS VARCHAR)

ELSE NULL

END AS Size

FROM

(

Select

\*

From

tempdb.sys.columns Where object\_id=OBJECT\_ID('tempdb.dbo.' + @TABLE\_NAME)

) A

JOIN sys.types B ON A.system\_type\_id = B.system\_type\_id AND B.name <> 'sysname'

) sub\_query

UNION ALL

SELECT

')'

,@NUM\_COLUMNS + 3

ORDER BY

column\_id

SELECT @COUNT = Count(\*) FROM #DECLARE

SET @I = 1

WHILE(@I <= @COUNT)

BEGIN

SELECT @STMT = CREATE\_STATEMENT FROM #DECLARE WHERE COLUMN\_ID = @I

print @STMT

SET @I = @I + 1

END

PRINT ' '

PRINT '-----------------------------------------------------------------------------'

PRINT ' '

DECLARE @MAX\_COLUMN\_TYPE\_LENGTH INT

INSERT INTO #SELECT\_DEFAULT\_DATA\_TYPE

SELECT

'SELECT'

, 1 AS column\_id

, null

UNION ALL

SELECT

CASE

WHEN Size IS NULL THEN

CASE column\_id

WHEN 1 THEN ' Cast(' + DefVal + ' as ' + COLUMN\_TYPE + ')'

ELSE ' , Cast(' + DefVal + ' as ' + COLUMN\_TYPE + ')'

END

ELSE

CASE column\_id

WHEN 1 THEN ' Cast(' + DefVal + ' as ' + COLUMN\_TYPE + '(' + CAST(size AS varchar) + ')' + ')'

ELSE ' , Cast(' + DefVal + ' as ' + COLUMN\_TYPE + '(' + CAST(size AS varchar) + ')' + ')'

END

END AS CREATE\_STATEMENT

,column\_id + 1 AS column\_id

, ' as ' + COLUMN\_NAME AS COLUMN\_NAME

FROM

(

SELECT

A.name AS COLUMN\_NAME

,B.name AS COLUMN\_TYPE

,A.column\_id

,CASE

WHEN B.name = 'nvarchar' THEN CASE WHEN A.max\_length = -1 THEN 'MAX' ELSE CAST(A.max\_length/2 AS VARCHAR) END

WHEN B.name = 'char' THEN CAST(A.max\_length AS VARCHAR)

WHEN B.name = 'varchar' THEN CASE WHEN A.max\_length = -1 THEN 'MAX' ELSE CAST(A.max\_length AS VARCHAR) END

WHEN B.name = 'varbinary' THEN CASE WHEN A.max\_length = -1 THEN 'MAX' ELSE CAST(A.max\_length AS VARCHAR) END

WHEN B.name = 'binary' THEN CAST(A.max\_length AS VARCHAR)

WHEN B.name = 'nchar' THEN CAST(A.max\_length/2 AS VARCHAR)

WHEN B.name = 'datetime2' THEN CAST(A.scale AS VARCHAR)

WHEN B.name = 'datetimeoffset' THEN CAST(A.scale AS VARCHAR)

WHEN B.name = 'time' THEN CAST(A.scale AS VARCHAR)

WHEN B.name = 'float' THEN CAST(A.precision AS VARCHAR)

WHEN B.name = 'decimal' THEN CAST(A.precision AS VARCHAR) + ',' + CAST(A.scale AS VARCHAR)

WHEN B.name = 'numeric' THEN CAST(A.precision AS VARCHAR) + ',' + CAST(A.scale AS VARCHAR)

ELSE NULL

END AS Size

,CASE

WHEN b.name in ('bigint', 'int', 'smallint', 'tinyint', 'bit', 'float', 'decimal', 'numeric') THEN '0'

WHEN b.name in ('datetime') THEN 'GetDate()'

WHEN b.name in ('time') THEN '''00:00'''

WHEN b.name in ('nvarchar', 'char', 'varchar', 'nchar') THEN ''''''

ELSE 'NULL'

END as DefVal

FROM

(

Select

\*

From

tempdb.sys.columns Where object\_id=OBJECT\_ID('tempdb.dbo.' + @TABLE\_NAME)

) A

JOIN sys.types B ON A.system\_type\_id = B.system\_type\_id AND B.name <> 'sysname'

) sub\_query

ORDER BY

column\_id

set @MAX\_COLUMN\_TYPE\_LENGTH = (SELECT MAX(LEN(CREATE\_STATEMENT)) FROM #SELECT\_DEFAULT\_DATA\_TYPE)

UPDATE #SELECT\_DEFAULT\_DATA\_TYPE

SET CREATE\_STATEMENT = CREATE\_STATEMENT + REPLICATE(' ', @MAX\_COLUMN\_TYPE\_LENGTH - LEN(CREATE\_STATEMENT)) + COALESCE(COLUMN\_NAME, '')

SELECT @COUNT = Count(\*) FROM #SELECT\_DEFAULT\_DATA\_TYPE

SET @I = 1

WHILE(@I <= @COUNT)

BEGIN

SELECT @STMT = CREATE\_STATEMENT FROM #SELECT\_DEFAULT\_DATA\_TYPE WHERE COLUMN\_ID = @I

print @STMT

SET @I = @I + 1

END

PRINT ' '

PRINT '-----------------------------------------------------------------------------'

PRINT ' '

INSERT INTO #SELECT\_NULL

SELECT

'SELECT'

,1 AS column\_id

, NULL

UNION ALL

SELECT

CASE

WHEN Size IS NULL THEN

CASE column\_id

WHEN 1 THEN ' Cast(' + DefVal + ' as ' + COLUMN\_TYPE + ')'

ELSE ' , Cast(' + DefVal + ' as ' + COLUMN\_TYPE + ')'

END

ELSE

CASE column\_id

WHEN 1 THEN ' Cast(' + DefVal + ' as ' + COLUMN\_TYPE + '(' + CAST(size AS varchar) + ')' + ')'

ELSE ' , Cast(' + DefVal + ' as ' + COLUMN\_TYPE + '(' + CAST(size AS varchar) + ')' + ')'

END

END AS CREATE\_STATEMENT

,column\_id + 1 AS column\_id

, ' as ' + COLUMN\_NAME AS COLUMN\_NAME

FROM

(

SELECT

A.name AS COLUMN\_NAME

,B.name AS COLUMN\_TYPE

,A.column\_id

,CASE

WHEN B.name = 'nvarchar' THEN CASE WHEN A.max\_length = -1 THEN 'MAX' ELSE CAST(A.max\_length/2 AS VARCHAR) END

WHEN B.name = 'char' THEN CAST(A.max\_length AS VARCHAR)

WHEN B.name = 'varchar' THEN CASE WHEN A.max\_length = -1 THEN 'MAX' ELSE CAST(A.max\_length AS VARCHAR) END

WHEN B.name = 'varbinary' THEN CASE WHEN A.max\_length = -1 THEN 'MAX' ELSE CAST(A.max\_length AS VARCHAR) END

WHEN B.name = 'binary' THEN CAST(A.max\_length AS VARCHAR)

WHEN B.name = 'nchar' THEN CAST(A.max\_length/2 AS VARCHAR)

WHEN B.name = 'datetime2' THEN CAST(A.scale AS VARCHAR)

WHEN B.name = 'datetimeoffset' THEN CAST(A.scale AS VARCHAR)

WHEN B.name = 'time' THEN CAST(A.scale AS VARCHAR)

WHEN B.name = 'float' THEN CAST(A.precision AS VARCHAR)

WHEN B.name = 'decimal' THEN CAST(A.precision AS VARCHAR) + ',' + CAST(A.scale AS VARCHAR)

WHEN B.name = 'numeric' THEN CAST(A.precision AS VARCHAR) + ',' + CAST(A.scale AS VARCHAR)

ELSE NULL

END AS Size

,'null' as DefVal

FROM

(

Select

\*

From

tempdb.sys.columns Where object\_id=OBJECT\_ID('tempdb.dbo.' + @TABLE\_NAME)

) A

JOIN sys.types B ON A.system\_type\_id = B.system\_type\_id AND B.name <> 'sysname'

) sub\_query

ORDER BY

column\_id

set @MAX\_COLUMN\_TYPE\_LENGTH = (SELECT MAX(LEN(CREATE\_STATEMENT)) FROM #SELECT\_NULL)

UPDATE #SELECT\_NULL

SET CREATE\_STATEMENT = CREATE\_STATEMENT + REPLICATE(' ', @MAX\_COLUMN\_TYPE\_LENGTH - LEN(CREATE\_STATEMENT)) + COALESCE(COLUMN\_NAME, '')

SELECT @COUNT = Count(\*) FROM #SELECT\_NULL

SET @I = 1

WHILE(@I <= @COUNT)

BEGIN

SELECT @STMT = CREATE\_STATEMENT FROM #SELECT\_NULL WHERE COLUMN\_ID = @I

print @STMT

SET @I = @I + 1

END

end

go