

# Replication Topology script

Last updated by | Holger Linke | Jan 24, 2023 at 12:28 AM PST

## Contents

- [Description](#)
- [Sample output](#)
- [Replication Topology.sql](#)

## Description

This script queries the Distribution database (and optionally the Publisher database, if the Distributor is also a Publisher) and draws out the replication topology. Its output helps with understanding the customer's replication environment and it displays various other details that might be relevant for troubleshooting the issue.

This script is useful if the customer's replication environment is very complex or if the customer is unable to describe the configuration in a meaningful way.

Let the customer run the script against their distribution database at the Distributor.

## Sample output

```
=====
REPLICATION TOPOLOGY
=====
SELECT THE PUBLICATION-SUBSCRIPTION PAIR FOR SCOPING THE CASE
=====

h1idist.eddd3e2e0e6c.database.windows.net (DISTRIBUTOR :: 12.0.2000 MI) (Distribution Database: Distribution)
|- PUBLISHERS
|---- repl_distributor (Publisher 1)
|   |- PUBLICATIONS
|   |---- Publication_Tran (Publication ID: 1; Publication type: Transactional (1-way); Publication databas
|   |   |- ARTICLES
|   |   |---- 1 article(s)
|   |   |- SUBSCRIPTIONS
|   |   |---- repl_distributor (Subscription Database: Repl_SUB; Subscription Type: Push)
```

## Replication Topology.sql

```
/** Start of Script **/
```

```
/******
```

```
PURPOSE : Display Replication Topology
```

```
VERSION : 1.0.0.1 -- Revised to support multiple distribution database scenario as well
```

```
RUNS ON : The Distributor Server; against the Distribution Database
```

```
Author: Suhas De (suhde@microsoft.com) & Gaurav Mathur (gmathur@microsoft.com)
```

```
This script queries the Distribution Database (and optionally the Publisher database, if the Distributor is also a Publisher) and draws out the replication topology. It helps us understand the replication topology and displays various other details that we need to understand the issue and thus helps us to troubleshoot the issue faster.
```

```
Currently we support only SQL Server 2005 and later instances.
```

```
*/
```

```
SET NOCOUNT ON
```

```
GO
```

```
IF ((SELECT COUNT(*) FROM TEMPDB.SYS.TABLES WHERE NAME = '##CE') > 0)
```

```
DROP TABLE ##CE
```

```
GO
```

```
CREATE TABLE ##CE ([DESCRIPTION] VARCHAR(500) NOT NULL, [VALUE] VARCHAR(500) NOT NULL)
```

```
GO
```

```
INSERT INTO ##CE VALUES('Continue', 1)
```

```
GO
```

```
DECLARE @CONSOLEMSG VARCHAR(1000)
```

```
DECLARE @SQLVersion VARCHAR(2)
```

```
SET @SQLVersion = CONVERT(VARCHAR(2), SERVERPROPERTY('ProductVersion'))
```

```
IF SUBSTRING(@SQLVersion, 2, 1) = '.'
```

```
SET @SQLVersion = SUBSTRING(@SQLVersion, 1, 1)
```

```
IF CONVERT(INT, @SQLVersion) < 9
```

```
BEGIN
```

```
SET @CONSOLEMSG=CONVERT(VARCHAR(24),GETDATE(),121)+ ' SQL Server connected to is not SQL Server 2005 or SQL Se
```

```
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
```

```
UPDATE ##CE SET [VALUE] = 0 WHERE [DESCRIPTION] = 'Continue'
```

```
END
```

```
GO
```

```
IF ((SELECT [VALUE] FROM ##CE WHERE [DESCRIPTION] = 'Continue') = 1)
```

```
BEGIN
```

```
DECLARE @CONSOLEMSG VARCHAR(1000)
```

```
DECLARE @DistInst VARCHAR(1)
```

```
DECLARE @BitEd VARCHAR(100)
```

```
DECLARE @xIndex INT
```

```
SELECT @DistInst = CONVERT(VARCHAR(1), ISNULL([IS_DISTRIBUTOR], 0)) FROM [MASTER].[SYS].[SERVERS] (NOLOCK) WHE
```

```
IF @DistInst IS NULL OR @DistInst = '0'
```

```
BEGIN
```

```
SET @CONSOLEMSG=CONVERT(VARCHAR(24),GETDATE(),121)+ ' Selected instance is not a distributor instance. Exiting
```

```
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
```

```
UPDATE ##CE SET [VALUE] = 0 WHERE [DESCRIPTION] = 'Continue'
```

```
END
```

```
ELSE
```

```
BEGIN
```

```
SET @CONSOLEMSG = REPLACE(CONVERT(VARCHAR(256), SERVERPROPERTY('ServerName'))) + ' (DISTRIBUTOR :: ' + CONVERT(
```

```
SELECT @BitEd = @@VERSION
```

```
SELECT @xIndex = CHARINDEX('X', UPPER(@BitEd))
```

```
SELECT @BitEd = SUBSTRING(UPPER(@BitEd), @xIndex, 3)
```

```
SET @CONSOLEMSG = @CONSOLEMSG + ' ' + @BitEd + ' )'
```

```
INSERT INTO ##CE VALUES('Distributor', @CONSOLEMSG)
```

```
END
```

```
END
```

```
GO
```

```
IF ((SELECT [VALUE] FROM ##CE WHERE [DESCRIPTION] = 'Continue') = 1)
```

```
BEGIN
```

```
DECLARE @CONSOLEMSG VARCHAR(1000)
```

```
SET @CONSOLEMSG = '====='
```

```
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
```

```
SET @CONSOLEMSG = ' REPLICATION TOPOLOGY'
```

```
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
```

```
SET @CONSOLEMSG = '====='
```

```
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
```

```

SET @CONSOLEMSG = ' SELECT THE PUBLICATION-SUBSCRIPTION PAIR FOR SCOPING THE CASE'
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
SET @CONSOLEMSG = '=====
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
SET @CONSOLEMSG = ' '
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
END
GO
IF ((SELECT [VALUE] FROM ##CE WHERE [DESCRIPTION] = 'Continue') = 1)
BEGIN
DECLARE @CONSOLEMSG VARCHAR(1000)
DECLARE @DISTRIBUTIONDBNAME SYSNAME
SET @DISTRIBUTIONDBNAME = '???'
DECLARE @CURRENTDATABASE SYSNAME
SELECT @CURRENTDATABASE = DB_NAME()
SELECT @DISTRIBUTIONDBNAME = NAME FROM SYS.DATABASES (NOLOCK) WHERE IS_DISTRIBUTOR = 1 AND NAME = @CURRENTDATA
SELECT @CONSOLEMSG = [VALUE] FROM ##CE WHERE [DESCRIPTION] = 'Distributor'
SET @CONSOLEMSG = @CONSOLEMSG + ' (Distribution Database: ' + @DISTRIBUTIONDBNAME + ' )'
DELETE ##CE WHERE [DESCRIPTION] = 'Distributor'
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
IF @CURRENTDATABASE <> @DISTRIBUTIONDBNAME
BEGIN
SET @CONSOLEMSG = ' Context Database is not the Distribution Database. Exiting.'
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
UPDATE ##CE SET [VALUE] = 0 WHERE [DESCRIPTION] = 'Continue'
END
END
GO
IF ((SELECT [VALUE] FROM ##CE WHERE [DESCRIPTION] = 'Continue') = 1)
BEGIN
DECLARE @CONSOLEMSG VARCHAR(1000)
DECLARE @DISTRIBUTORSERVERNAME SYSNAME
DECLARE @PUBLISHERNAME SYSNAME
DECLARE @PUBLISHERID INT
DECLARE @PUBLISHERNUMBER INT
DECLARE @PUBLICATIONNAME SYSNAME
DECLARE @PUBLICATIONID INT
DECLARE @PUBLICATIONTYPE INT
DECLARE @PUBLICATIONDATABASE SYSNAME
DECLARE @ALLOW_QUEUED_TRAN INT
DECLARE @STMT VARCHAR(MAX)
DECLARE @NUMARTICLES INT
DECLARE @RESERVEDSIZE BIGINT
DECLARE @USED SIZE BIGINT
DECLARE @INDEXSIZE BIGINT
DECLARE @SUBSCRIBERNAME SYSNAME
DECLARE @SUBSCRIPTIONDB SYSNAME
DECLARE @SUBSCRIPTIONTYPE INT
SET @PUBLISHERNUMBER = 0
SET @DISTRIBUTORSERVERNAME = CONVERT(SYSNAME, SERVERPROPERTY('ServerName'))
SET @CONSOLEMSG = ' |- PUBLISHERS'
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
DECLARE PUBLISHERCURSOR CURSOR LOCAL READ_ONLY FOR
SELECT DISTINCT S.NAME , PUB.PUBLISHER_ID FROM SYS.SERVERS (NOLOCK) S JOIN DBO.MSPUBLICATIONS (NOLOCK) PUB
ON S.SERVER_ID = PUB.PUBLISHER_ID
OPEN PUBLISHERCURSOR
FETCH NEXT FROM PUBLISHERCURSOR INTO @PUBLISHERNAME, @PUBLISHERID
WHILE @@FETCH_STATUS = 0
BEGIN
SET @PUBLISHERNUMBER = @PUBLISHERNUMBER + 1
SET @CONSOLEMSG = ' |---- ' + @PUBLISHERNAME + ' (Publisher ' + CONVERT(VARCHAR(10), @PUBLISHERNUMBER) + ' )'
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
SET @CONSOLEMSG = ' |- PUBLICATIONS'
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
DECLARE PUBLICATIONCURSOR CURSOR LOCAL READ_ONLY FOR
SELECT PUBLICATION, PUBLICATION_ID, PUBLICATION_TYPE, PUBLISHER_DB, ALLOW_QUEUED_TRAN
FROM DBO.MSPUBLICATIONS (NOLOCK) WHERE PUBLISHER_ID = @PUBLISHERID
OPEN PUBLICATIONCURSOR
FETCH NEXT FROM PUBLICATIONCURSOR INTO @PUBLICATIONNAME, @PUBLICATIONID,
@PUBLICATIONTYPE, @PUBLICATIONDATABASE, @ALLOW_QUEUED_TRAN

```

```

WHILE @@FETCH_STATUS = 0
BEGIN
SET @CONSOLEMSG = '          |---- ' + @PUBLICATIONNAME + ' ('
SET @CONSOLEMSG = @CONSOLEMSG + 'Publication ID: ' + CONVERT(VARCHAR(10), @PUBLICATIONID) + '; '
IF @PUBLICATIONTYPE = 0
BEGIN
IF @ALLOW_QUEUED_TRAN = 0
SET @CONSOLEMSG = @CONSOLEMSG + 'Publication type: Transactional (1-way); '
ELSE
SET @CONSOLEMSG = @CONSOLEMSG + 'Publication type: Transactional (2-way); '
END
ELSE IF @PUBLICATIONTYPE = 1
SET @CONSOLEMSG = @CONSOLEMSG + 'Publication type: Snapshot; '
ELSE IF @PUBLICATIONTYPE = 2
SET @CONSOLEMSG = @CONSOLEMSG + 'Publication type: Merge; '
SET @CONSOLEMSG = @CONSOLEMSG + 'Publication database: ' + @PUBLICATIONDATABASE + ')'
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
SET @CONSOLEMSG = 'XXX'
IF @PUBLICATIONTYPE < 2
BEGIN
SET @CONSOLEMSG = '          |- ARTICLES'
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
SELECT @NUMARTICLES = COUNT(ARTICLE_ID) FROM MSARTICLES (NOLOCK) WHERE PUBLICATION_ID = @PUBLICATIONID AND PUB
SET @CONSOLEMSG = '          |---- ' + CONVERT(VARCHAR(10), @NUMARTICLES) + ' article(s)'
END
ELSE
BEGIN
IF @DISTRIBUTORSERVERNAME = @PUBLISHERNAME
BEGIN
SET @CONSOLEMSG = '          |- ARTICLES'
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
SET @STMT = 'SET NOCOUNT ON' + CHAR(13)
SET @STMT = @STMT + 'DECLARE @NUMART INT' + CHAR(13)
SET @STMT = @STMT + 'SELECT @NUMART = COUNT(MA.OBJID) FROM ' + @PUBLICATIONDATABASE + '.DBO.SYSMERGEARTICLES ('
SET @STMT = @STMT + 'INSERT INTO ##CE VALUES (''NUMART'', @NUMART)' + CHAR(13)
EXEC (@STMT)
SELECT @NUMARTICLES = [VALUE] FROM ##CE WHERE [DESCRIPTION] = 'NUMART'
DELETE ##CE WHERE [DESCRIPTION] = 'NUMART'
--
SET @CONSOLEMSG = '          |----- ' + CONVERT(VARCHAR(10), @NUMARTICLES)
SET @CONSOLEMSG = '          |---- ' + CONVERT(VARCHAR(10), @NUMARTICLES) + ' article(s)'
END
END
IF @DISTRIBUTORSERVERNAME = @PUBLISHERNAME
BEGIN
IF @PUBLICATIONTYPE < 2
BEGIN
SET @STMT = 'SET NOCOUNT ON' + CHAR(13)
SET @STMT = @STMT + 'DECLARE @Reserved BIGINT' + CHAR(13)
SET @STMT = @STMT + 'DECLARE @Used BIGINT' + CHAR(13)
SET @STMT = @STMT + 'DECLARE @Index BIGINT' + CHAR(13)
SET @STMT = @STMT + 'SELECT @Reserved = SUM([Reserved Size (KB)]),' + CHAR(13)
SET @STMT = @STMT + '@Used = SUM([Used Size (KB)]),' + CHAR(13)
SET @STMT = @STMT + '@Index = SUM([Index Size (KB)])' + CHAR(13)
SET @STMT = @STMT + 'FROM (SELECT SUM([PS].[Reserved_Page_Count]) * 8 AS [Reserved Size (KB)],' + CHAR(13)
SET @STMT = @STMT + 'SUM([PS].[Used_Page_Count]) * 8 AS [Used Size (KB)],' + CHAR(13)
SET @STMT = @STMT + 'SUM(' + CHAR(13)
SET @STMT = @STMT + 'CASE' + CHAR(13)
SET @STMT = @STMT + 'WHEN ([PS].[index_id] < 2) THEN ([PS].[in_row_data_page_count] + [PS].[lob_used_page_cou'
SET @STMT = @STMT + 'ELSE [PS].[lob_used_page_count] + [PS].[row_overflow_used_page_count]' + CHAR(13)
SET @STMT = @STMT + 'END' + CHAR(13)
SET @STMT = @STMT + ') * 8 AS [Index Size (KB)]' + CHAR(13)
SET @STMT = @STMT + 'FROM [MSArticles] [MA] (NOLOCK)' + CHAR(13)
SET @STMT = @STMT + 'JOIN ' + @PUBLICATIONDATABASE + '.DBO.[SysArticles] [SA] (NOLOCK)' + CHAR(13)
SET @STMT = @STMT + 'ON [SA].[artid] = [MA].[article_id]' + CHAR(13)
SET @STMT = @STMT + 'JOIN ' + @PUBLICATIONDATABASE + '.[sys].[dm_db_Partition_Stats] [PS] (NOLOCK)' + CHAR(13)
SET @STMT = @STMT + 'ON [PS].[object_id] = [SA].[objid]' + CHAR(13)
SET @STMT = @STMT + 'WHERE [MA].[publisher_id] = ' + CONVERT(VARCHAR(10), @PUBLISHERID) + CHAR(13)
SET @STMT = @STMT + 'AND [MA].[publication_id] = ' + CONVERT(VARCHAR(10), @PUBLICATIONID) + CHAR(13)
SET @STMT = @STMT + 'GROUP BY [SA].[objid], [MA].[source_owner], [MA].[article]] A' + CHAR(13)
SET @STMT = @STMT + 'INSERT INTO ##CE VALUES (''Reserved'', @Reserved)' + CHAR(13)

```

```

SET @STMT = @STMT + 'INSERT INTO ##CE VALUES (''Used'', @Used)' + CHAR(13)
SET @STMT = @STMT + 'INSERT INTO ##CE VALUES (''Index'', @Index)' + CHAR(13)
EXEC (@STMT)
SELECT @RESERVEDSIZE = [VALUE] FROM ##CE WHERE [DESCRIPTION] = 'Reserved'
SELECT @USEDSize = [VALUE] FROM ##CE WHERE [DESCRIPTION] = 'Used'
SELECT @INDEXSize = [VALUE] FROM ##CE WHERE [DESCRIPTION] = 'Index'
SET @CONSOLEMSG = @CONSOLEMSG + '; Reserved Space = ' + CONVERT(VARCHAR(20), @RESERVEDSIZE) + ' KB, '
SET @CONSOLEMSG = @CONSOLEMSG + 'Used Space = ' + CONVERT(VARCHAR(20), @USEDSize) + ' KB, '
SET @CONSOLEMSG = @CONSOLEMSG + 'Index Space = ' + CONVERT(VARCHAR(20), @INDEXSize) + ' KB'
DELETE ##CE WHERE [DESCRIPTION] IN ('Reserved', 'Used', 'Index')
END
ELSE
BEGIN
SET @STMT = 'SET NOCOUNT ON' + CHAR(13)
SET @STMT = @STMT + 'DECLARE @Reserved BIGINT' + CHAR(13)
SET @STMT = @STMT + 'DECLARE @Used BIGINT' + CHAR(13)
SET @STMT = @STMT + 'DECLARE @Index BIGINT' + CHAR(13)
SET @STMT = @STMT + 'SELECT @Reserved = SUM([Reserved Size (KB)]),' + CHAR(13)
SET @STMT = @STMT + '@Used = SUM([Used Size (KB)]),' + CHAR(13)
SET @STMT = @STMT + '@Index = SUM([Index Size (KB)])' + CHAR(13)
SET @STMT = @STMT + 'FROM (SELECT SUM([PS].[Reserved_Page_Count]) * 8 AS [Reserved Size (KB)],' + CHAR(13)
SET @STMT = @STMT + 'SUM([PS].[Used_Page_Count]) * 8 AS [Used Size (KB)],' + CHAR(13)
SET @STMT = @STMT + 'SUM(' + CHAR(13)
SET @STMT = @STMT + 'CASE' + CHAR(13)
SET @STMT = @STMT + ' WHEN ([PS].[index_id] < 2) THEN ([PS].[in_row_data_page_count] + [PS].[lob_used_page_cou'
SET @STMT = @STMT + ' ELSE [PS].[lob_used_page_count] + [PS].[row_overflow_used_page_count]' + CHAR(13)
SET @STMT = @STMT + ' END' + CHAR(13)
SET @STMT = @STMT + ') * 8 AS [Index Size (KB)]' + CHAR(13)
SET @STMT = @STMT + 'FROM ' + @PUBLICATIONDATABASE + '.DBO.SYSMERGEARTICLES MA (NOLOCK) JOIN ' + @PUBLICATIOND
SET @STMT = @STMT + 'JOIN ' + @PUBLICATIONDATABASE + '.[sys].[dm_db_Partition_Stats] [PS] (NOLOCK) ON [PS].[ob
SET @STMT = @STMT + 'WHERE MP.PUBLISHER_DB = '' + @PUBLICATIONDATABASE + '' AND MP.NAME = '' + @PUBLICATION
SET @STMT = @STMT + 'INSERT INTO ##CE VALUES (''Reserved'', @Reserved)' + CHAR(13)
SET @STMT = @STMT + 'INSERT INTO ##CE VALUES (''Used'', @Used)' + CHAR(13)
SET @STMT = @STMT + 'INSERT INTO ##CE VALUES (''Index'', @Index)' + CHAR(13)
EXEC (@STMT)
SELECT @RESERVEDSIZE = [VALUE] FROM ##CE WHERE [DESCRIPTION] = 'Reserved'
SELECT @USEDSize = [VALUE] FROM ##CE WHERE [DESCRIPTION] = 'Used'
SELECT @INDEXSize = [VALUE] FROM ##CE WHERE [DESCRIPTION] = 'Index'
SET @CONSOLEMSG = @CONSOLEMSG + '; Reserved Space = ' + CONVERT(VARCHAR(20), @RESERVEDSIZE) + ' KB, '
SET @CONSOLEMSG = @CONSOLEMSG + 'Used Space = ' + CONVERT(VARCHAR(20), @USEDSize) + ' KB, '
SET @CONSOLEMSG = @CONSOLEMSG + 'Index Space = ' + CONVERT(VARCHAR(20), @INDEXSize) + ' KB'
DELETE ##CE WHERE [DESCRIPTION] IN ('Reserved', 'Used', 'Index')
END
END
IF @CONSOLEMSG <> 'XXX'
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
SET @CONSOLEMSG = ' | - SUBSCRIPTIONS'
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
IF @PUBLICATIONTYPE < 2
BEGIN
DECLARE SUBSCRIPTIONCURSOR CURSOR LOCAL READ_ONLY FOR
SELECT DISTINCT S.NAME, SUB.SUBSCRIBER_DB, SUB.SUBSCRIPTION_TYPE
FROM SYS.SERVERS S (NOLOCK) JOIN MSSUBSCRIPTIONS SUB (NOLOCK) ON S.SERVER_ID = SUB.SUBSCRIBER_ID
WHERE SUB.PUBLICATION_ID = @PUBLICATIONID AND SUB.PUBLISHER_DB = @PUBLICATIONDATABASE AND SUB.SUBSCRIBER_ID >=
OPEN SUBSCRIPTIONCURSOR
FETCH NEXT FROM SUBSCRIPTIONCURSOR INTO @SUBSCRIBERNAME, @SUBSCRIPTIONDB, @SUBSCRIPTIONTYPE
WHILE @@FETCH_STATUS = 0
BEGIN
SET @CONSOLEMSG = ' | ---- ' + @SUBSCRIBERNAME + ' ('
SET @CONSOLEMSG = @CONSOLEMSG + 'Subscription Database: ' + @SUBSCRIPTIONDB + '; '
IF @SUBSCRIPTIONTYPE = 0
SET @CONSOLEMSG = @CONSOLEMSG + 'Subscription Type: Push)'
ELSE IF @SUBSCRIPTIONTYPE = 1
SET @CONSOLEMSG = @CONSOLEMSG + 'Subscription Type: Pull)'
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
FETCH NEXT FROM SUBSCRIPTIONCURSOR INTO @SUBSCRIBERNAME, @SUBSCRIPTIONDB, @SUBSCRIPTIONTYPE
END
CLOSE SUBSCRIPTIONCURSOR
DEALLOCATE SUBSCRIPTIONCURSOR
END

```

```

ELSE
BEGIN
DECLARE SUBSCRIPTIONCURSOR CURSOR LOCAL READ_ONLY FOR
SELECT SUBSCRIBER, SUBSCRIBER_DB, SUBSCRIPTION_TYPE
FROM MSMERGE_SUBSCRIPTIONS (NOLOCK) WHERE PUBLICATION_ID = @PUBLICATIONID AND PUBLISHER_DB = @PUBLICATIONDATAB
OPEN SUBSCRIPTIONCURSOR
FETCH NEXT FROM SUBSCRIPTIONCURSOR INTO @SUBSCRIBERNAME, @SUBSCRIPTIONDB, @SUBSCRIPTIONTYPE
WHILE @@FETCH_STATUS = 0
BEGIN
SET @CONSOLEMSG = '          |---- ' + @SUBSCRIBERNAME + ' ('
SET @CONSOLEMSG = @CONSOLEMSG + 'Subscription Database: ' + @SUBSCRIPTIONDB + '; '
IF @SUBSCRIPTIONTYPE = 0
SET @CONSOLEMSG = @CONSOLEMSG + 'Subscription Type: Push)'
ELSE IF @SUBSCRIPTIONTYPE = 1
SET @CONSOLEMSG = @CONSOLEMSG + 'Subscription Type: Pull)'
RAISERROR (@CONSOLEMSG,10,1) WITH NOWAIT
FETCH NEXT FROM SUBSCRIPTIONCURSOR INTO @SUBSCRIBERNAME, @SUBSCRIPTIONDB, @SUBSCRIPTIONTYPE
END
CLOSE SUBSCRIPTIONCURSOR
DEALLOCATE SUBSCRIPTIONCURSOR
END
FETCH NEXT FROM PUBLICATIONCURSOR INTO @PUBLICATIONNAME, @PUBLICATIONID, @PUBLICATIONTYPE, @PUBLICATIONDATABASE
END
CLOSE PUBLICATIONCURSOR
DEALLOCATE PUBLICATIONCURSOR
FETCH NEXT FROM PUBLISHERCURSOR INTO @PUBLISHERNAME, @PUBLISHERID
END
CLOSE PUBLISHERCURSOR
DEALLOCATE PUBLISHERCURSOR
END
GO
DROP TABLE ##CE
GO

```

/\*\* End of Script \*\*/``

<br>

**\*\*How good have you found this content?\*\***

```

<div>
<span>
<a href='https://csswikifedback.azurewebsites.net/feedback/smile?wiki=AzureSQLDB&pageId=286489&pageTitle=Repl
<img alt='Smile' src='https://sqldbwikifedback.blob.core.windows.net/assets/smile.png'>
</a>
</span>

<span>
<a href='https://csswikifedback.azurewebsites.net/feedback/frown?wiki=AzureSQLDB&pageId=286489&pageTitle=Repl
<img alt='Sad' src='https://sqldbwikifedback.blob.core.windows.net/assets/sad.png'>
</a>
</span>
</div>

```