

## How to configure replication using snapshot

To configure the transactional replication using snapshot, we need to configure the distributor, Publisher and Subscriber. Also we need to handshake each server with another one. Below are the detailed steps for configuring replication from scratch.

**Note: Change the variables and location according to env.**

Lets assume below are the server details and default instance of SQL server is installed on each server.

1. SQLNodeDist = Distributor Server
2. SQLNodeSubs = Subscriber Server
3. SQLNodePubl = Publisher Server

### Step 1: Configure Distributor

Connect SSMS on Distributor Server and execute below command

```
USE MASTER
GO
--Declaring variables
DECLARE @DistributorServer SYSNAME, @DistributorAdminDBPassword VARCHAR(60)
SELECT @DistributorServer = 'SQLNodeDist'
SELECT @DistributorAdminDBPassword = 'Test123$'
--Set the name and then print the name of the distributor.
DECLARE @DistributorServer_msg varchar (50)
SET @DistributorServer_msg = ' The name of the distributor server is '
PRINT char (10) + char (13) + @DistributorServer_MSG + @DistributorServer + Char (10)
+ Char (13) SELECT @DistributorServer as DistributionServer, @DistributorAdminDBPassw
ord as DistributorAdminDBPassword
EXEC SP_ADDDISTRIBUTOR @distributor = @DistributorServer, @password = @DistributorA
dminDBPassword
GO
```

### Step 2: Add Distribution database on distributor server

Connect SSMS on Distributor Server and execute below command

```
USE MASTER
GO
DECLARE @DistributorAdminDBPassword varchar(60)
DECLARE @dataFolder varchar(1000)
DECLARE @logFolder varchar(1000)
DECLARE @dataFileSize INT
DECLARE @LOGFileSize INT
SELECT @DistributorAdminDBPassword = 'Test123$'
SELECT @dataFolder = 'E:\SQLDATA\'
SELECT @logFolder = 'F:\SQLLOG\'
SELECT @dataFileSize = '200'
SELECT @LOGFileSize = '10'
SELECT 'Configuration values are',@DistributorAdminDBPassword
DistributorAdminDBPassword ,@dataFolder dataFolder, @logFolder
logFolder , @dataFileSize dataFileSize ,@LOGFileSize LOGFileSize
--needs to be change as per ENV
EXEC SP_ADDDISTRIBUTIONDB
@database = N'distribution'
```

```
,@data_folder = @dataFolder
,@data_file = N'distribution.MDF'
,@data_file_size = @dataFileSize
,@log_folder = @logFolder
,@log_file = N'distribution.LDF'
,@log_File_Size = @LOGFileSize
,@min_distretention = 0
,@max_distretention = 72
,@history_retention = 48
,@security_mode = 1
--,@login = @DistributorAgent
,@Password = @DistributorAdminDBPassword
GO
```

Step 3: Add Remote Publishers(Telling the Distributor server that these are your possible publishers)

Connect SSMS on Distributor Server and execute below command

```
DECLARE @PublisherServer sysname,@AltSnapshotFolder varchar(600)
SELECT @PublisherServer = 'SQLNodePubl'
SELECT @AltSnapshotFolder = '\\SQLNodedist\ReplSnapshot\'
SELECT @PublisherServer PublisherServer ,@AltSnapshotFolder AltSnapshotFolder
EXEC sp_adddistpublisher @publisher = @PublisherServer, @distribution_db = N'distribution', @security_mode = 1, @working_directory = @AltSnapshotFolder
,@trusted = N'false', @thirdparty_flag = 0, @publisher_type = N'MSSQLSERVER'
GO
```

Step 4: Add Remote Distributor on Publisher server(Telling the publisher server that this is your distributor)

Connect SSMS on Publisher Server and execute below command

```
USE MASTER
DECLARE @DistributorServer VARCHAR(500)
SELECT @DistributorServer = 'SQLNodeDist'
DECLARE @DistributorAdminPassword varchar (500)
SELECT @DistributorAdminPassword = 'Test123$'
SELECT @DistributorAdminPassword As DistributorAdminPassword , @DistributorServer AS DistributorServer
EXEC sp_adddistributor @distributor = @DistributorServer, @password = @DistributorAdminPassword
GO
```

Step 5: Add Remote Distributor on Publisher server(Telling the publisher server that this is your distributor)

Connect SSMS on Publisher Server and execute below command

```
USE MASTER
DECLARE @SubscriberServer varchar(500)
SELECT @SubscriberServer = 'SQLNodeSubs'
SELECT @SubscriberServer as SubscriberServer
EXEC SP_ADDSUBSCRIBER @subscriber = @SubscriberServer, @type = 0, @description = N''
GO
```

**Initial setup of relationship between Distributor, Publisher and Subscriber is complete.**

Step 6: Add Pull subscriptions.

Connect SSMS on Subscriber Server and execute below command

```
----- Adding the Subscribers. Connect to the first Subscriber first, execute it.
Repeat the same process for the second Subscriber again.
USE Reporting_Subscriber
GO
```

```

EXEC SP_ADDPULLSUBSCRIPTION @publisher = 'SQLNodePubl'
, @publication = 'Trans_Publication_PublisherDB'
, @publisher_db = 'Publisherdb' , @independent_agent = N'True'
, @subscription_type = N'pull', @description = N'', @update_mode = N'read only'
, @immediate_sync = 0
--- Adding the job for the subscriber.
EXEC sp_addpullsubscription_agent @publisher = 'SQLNodePubl', @publisher_db = 'PublisherDB', @publication = 'Trans_Publication_PublisherDB', @distributor = 'SQLNodeDist' ,
@distributor_security_mode = 1, @distributor_login = N'', @distributor_password = N'',
@enabled_for_syncmgr = N'False', @frequency_type = 64, @frequency_interval = 0, @frequency_relative_interval = 0, @frequency_recurrence_factor = 0, @frequency_subday = 0,
@frequency_subday_interval = 0, @active_start_time_of_day = 0, @active_end_time_of_day
= 235959, @active_start_date = 20150627, @active_end_date = 99991231, @alt_snapshot_folder = '\\SQLNodeDist\ReplSnapshot\ ', @working_directory = N'', @use_ftp = N'False',
@job_login = null, @job_password = null, @publication_type = 0
GO

```

## Step 7: Create publications

Connect SSMS on Publisher Server and execute below command

```

USE PublisherDB
GO
EXEC sp_replicationdboption @dbname = 'PublisherDB'
, @optname = 'Publish'
, @value = 'true'
GO
exec [Publisherdb].sys.sp_addlogreader_agent @job_login = null, @job_password = null,
@publisher_security_mode = 1
GO
exec [Publisherdb].sys.sp_addqreader_agent @job_login = null, @job_password = null, @frompublisher = 1
GO
---- Adding the transactional publication
USE Publisher
GO
EXEC sp_addpublication @publication = 'Trans_Publication_PublisherDB'
, @description = N'Transactional publication'
, @sync_method = N'concurrent'
, @retention = 0, @allow_push = N'true', @allow_pull = N'true'
, @allow_anonymous = N'false', @enabled_for_internet = N'false'
, @snapshot_in_defaultfolder = N'false'
, @alt_snapshot_folder = '\\SQLNodeDist\ReplSnapshot\'
, @compress_snapshot = N'false', @ftp_port = 21
, @ftp_login = N'anonymous'
, @allow_subscription_copy = N'false'
, @add_to_active_directory = N'false', @repl_freq = N'continuous'
, @status = N'active', @independent_agent = N'true'
, @immediate_sync = N'false', @allow_sync_tran = N'false'
, @autogen_sync_procs = N'false'
, @allow_queued_tran = N'false', @allow_dts = N'false'
, @replicate_ddl = 1, @allow_initialize_from_backup = N'false'
, @enabled_for_p2p = N'false', @enabled_for_het_sub = N'false'
GO
EXEC SP_ADDPUBLICATION_SNAPSHOT
@publication = 'Trans_Publication_PublisherDB'
, @frequency_type = 4
, @frequency_interval = 1
, @frequency_relative_interval = 0, @frequency_recurrence_factor = 0

```

```

, @frequency_subday = 8, @frequency_subday_interval = 4, @active_start_t
ime_of_day = 0
, @active_end_time_of_day = 235959, @active_start_date = 0, @active_end_
date = 0
, @job_login = null, @job_password = null
, @publisher_security_mode = 1

```

*Publication has been setup now*

*THE NEXT STEP IS TO ADD AN ARTICLE TO THE PUBLICATION and after adding an article we need to run the sp\_addsubscription proc on the Publisher*

### Some ground work to limit the snapshot generation

```

use Publisherdb
DECLARE @DbName varchar(500)
DECLARE @PublicationName varchar(500)
SELECT @DbName = DB_NAME()
SELECT @PublicationName = name from syspublications
PRINT 'Adding article on the database ' + @DbName + char(10) + char(13)
PRINT 'Publication name is ' + @PublicationName + char(10) + char(13)
/*This will ensure that the snapshot will run only for the
   tables added and not the whole snapshot.*/
EXEC sp_changepublication
@publication = @PublicationName,
@property = 'allow_anonymous' ,
@value = 'false'
EXEC sp_changepublication
@publication = @PublicationName,
@property = 'immediate_sync' ,
@value = 'false'

```

### Step 8: Add article

Connect SSMS on Publisher Server and execute below command

```

use [Publisherdb]
exec sp_addarticle @publication = N'Trans_Publication_PublisherDB', @article = N'UserP
asswords', @source_owner = N'dbo', @source_object = N'UserPasswords', @type = N'logbas
ed', @description = N'', @creation_script = N'', @pre_creation_cmd = N'drop', @schema_
option = 0x000000000080350DF, @identityrangemanagementoption = N'none', @destination_ta
ble = N'UserPasswords', @destination_owner = N'dbo', @status = 24, @vertical_partition
= N'false', @ins_cmd = N'CALL [dbo].[sp_MSins_dboUserPasswords]', @del_cmd = N'CALL
[dbo].[sp_MSdel_dboUserPasswords]', @upd_cmd = N'SCALL
[dbo].[sp_MSupd_dboUserPasswords]'
GO

```

```

use [Publisherdb]
GO
EXEC SP_ADDSUBSCRIPTION @publication = 'Trans_Publication_PublisherDB', @subscriber =
'SQLNodeSubs', @destination_db = 'Reporting_Subscriber', @subscription_type = N'Pull',
@sync_type = N'Automatic', @article = N'ALL', @update_mode = N'read
only', @subscriber_type = 0
--EXEC SP_REFRESHSUBSCRIPTIONS @publication = N'Trans_Publication_PublisherDB'

```

### Step 8: Start the snapshot agent job.

Once we added the article, start the snapshot agent job. Also check the Log reader and Distribution agent job, and their schedule. Adjust the schedule according to environment.

Step 9: Once the previous snapshot delivered, we can add more article to the publication and now this time after adding article we only need to refresh the subscription as it is already added in first time.

```
use [Publisherdb]
exec sp_addarticle @publication = N'Trans_Publication_PublisherDB', @article = N'UserP
asswords', @source_owner = N'dbo', @source_object = N'UserPasswords', @type = N'logbas
ed', @description = N'', @creation_script = N'', @pre_creation_cmd = N'drop', @schema_
option = 0x000000000080350DF, @identityrangemanagementoption = N'none', @destination_ta
ble = N'UserPasswords', @destination_owner = N'dbo', @status = 24, @vertical_partition
= N'false', @ins_cmd = N'CALL [dbo].[sp_MSins_dboUserPasswords]', @del_cmd = N'CALL
[dbo].[sp_MSdel_dboUserPasswords]', @upd_cmd = N'SCALL
[dbo].[sp_MSupd_dboUserPasswords]'
GO
EXEC SP_REFRESHSUBSCRIPTIONS @publication = N'Trans_Publication_PublisherDB'
```