

Bidirectional Transactional Replication

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:30 AM PST

Contents

- [Bidirectional Transaction Replication sample script](#)
 - [Introduction](#)
 - [Prerequisites:](#)
 - [Section 1 - Preparing the Managed Instances for Transacti...](#)
 - [Section 2 - Create the initial publication on Managed Insta...](#)
 - [Section 3 - Create the publication on Managed Instance 2 ...](#)
 - [References](#)

Bidirectional Transactional Replication in SQL Managed Instance

This article is also published in our public blog at [Bidirectional Transactional Replication in SQL Managed Instance](#) .

Its content can be shared freely with customer.

Bidirectional Transaction Replication sample script

This article provides you with a sample script for creating a Bidirectional Transactional Replication between two Managed Instances. It keeps details as simple and straightforward as possible to demonstrate the replication feature itself, and to allow for a quick setup to test specific functionality. We are using this type of script in our support work to reproduce customer issues and test the behaviour of specific configuration options.

Refer to article [Considerations on Bidirectional Transactional Replication](#) for a discussion about the advantages, risks, and potential pitfalls of bidirectional transactional replication.

Introduction

Bidirectional transactional replication is a specific Transactional Replication topology that allows two SQL Server instances or databases to replicate changes to each other. Each of the two databases publishes data and then subscribes to a publication with the same data from the other database. The "@loopback_detection" feature ensures that changes are only sent to the Subscriber and do not result in the changes being sent back to the Publisher.

The databases that are providing the publication/subscription pairs can be hosted either on the same SQL instance or on two different SQL instances. The SQL instances can either be SQL Server on-premise, SQL Server hosted in a Virtual Machine, SQL Managed Instance on Azure, or a combination of each. You just have to make sure that the instances can connect to each other. If you add a subscription by using the fully-qualified domain name (FQDN), verify that the server name (@@SERVERNAME) of the Subscriber returns the FQDN. If the Subscriber server name does not return the FQDN, changes that originate from that Subscriber may cause primary key violations.

Prerequisites:

- This article assumes that you are already familiar with Transactional Replication on Managed Instance.
- It assumes that you know the basics of Transactional Replication and how to configure Transactional Replication.
- You need to have an Azure Storage file share and its account key available for the snapshot files, same as you need for a simple, one-directional Transactional Replication.
- You need two Managed Instances located in the same Azure Virtual Network (VNet), to minimize the risk of connectivity issues.
- The server name (@@SERVERNAME) of the Managed Instances return their fully-qualified domain name (FQDN).

Section 1 - Preparing the Managed Instances for Transactional Replication

Run this section at both Managed Instances before moving on to section 2. Each Managed Instance will be configured as its own Publisher and Distributor, and each Managed Instance will become a Subscriber to the other Managed Instance.

This section of the script includes the following steps:

- adds a sysadmin login and user to the Managed Instance
- enables Distribution on the Managed Instance
- creates a database and marks it as published for Transactional Replication
- configures the security settings for the Log Reader Agent.

```

/*****
-- Bidirectional Transaction Replication sample script
-- needs 2 Managed Instances located in the same VNet
-- each MI will be configured as Publisher, Distributor and Subscriber
*****/

/*****
/**** SECTION 1 ****/
/**** run this section at both Managed Instances before moving to section 2 ****/
*****/

-- Create the SQL Login for the replication jobs and agents
SET NOCOUNT ON
GO
USE [master]
GO
CREATE LOGIN TRANREPLADMIN WITH PASSWORD = '$trongPa11word';
GO
CREATE USER TRANREPLADMIN FROM LOGIN TRANREPLADMIN
GO

ALTER SERVER ROLE [sysadmin] ADD MEMBER [TRANREPLADMIN]
/* Required - otherwise the Log Reader agent might fail on startup with:
The process could not execute 'sp_MSadd_replcmds' on '<MI name>.database.windows.net'.
Only members of the sysadmin fixed server role can perform this operation.'.
Batches were not committed to the Distributor.
*/
GO

-- Add the distributor and the distribution database
USE [master]
GO
-- check if distribution has already been configured:
exec sp_get_distributor
GO
-- check server name to verify to be connected to the correct server
select @@servername
GO

EXEC sp_adddistributor = @@servername, @password = N'$trongPa11word';
GO
EXEC sp_adddistributiondb @database = N'Distribution';
GO

-- Configure the Publisher at the Distributor
-- requires an Azure file share and a valid account key for it
-- CRITICAL: Port 445 (TCP outbound) needs to be open in the security rules of the Managed Instance subnet to
EXEC sp_adddistpublisher @publisher = @@servername,
    @distribution_db = N'Distribution',
    @security_mode = 0,
    @login = N'TRANREPLADMIN',
    @password = N'$trongPa11word',
    @working_directory = N'\\storagename.file.core.windows.net\snapshots',
    @storage_connection_string = N'DefaultEndpointsProtocol=https;AccountName=storagename;AccountKey=9vNdOFJ0
GO

-- Configure the SQL user for the distribution database
USE [Distribution]
GO
CREATE USER TRANREPLADMIN FROM LOGIN TRANREPLADMIN
GO
EXEC sp_addrolemember N'db_owner', N'TRANREPLADMIN'
GO
USE [master]
GO

-- confirm that distribution has been configured:
exec sp_get_distributor

```

```

GO

-- Create the replication database
SET NOCOUNT ON
GO
USE [master]
GO
CREATE DATABASE [replicationdb]
GO

USE [replicationdb]
GO
CREATE USER TRANREPLADMIN FROM LOGIN TRANREPLADMIN
GO
EXEC sp_addrolemember N'db_owner', N'TRANREPLADMIN'
GO
USE [master]
GO

-- Enable the replication database for publication
USE [master]
GO
exec sp_replicationdboption @dbname = N'replicationdb', @optname = N'publish', @value = N'true'
GO

-- configure the Log Reader agent
exec [replicationdb].sys.sp_addlogreader_agent
    @job_login = 'TRANREPLADMIN', @job_password = '$trongPa11word',
    @publisher_security_mode = 0,
    @publisher_login = 'TRANREPLADMIN', @publisher_password = '$trongPa11word'
GO

/**** END OF SECTION 1 *****/

```

Section 2 - Create the initial publication on Managed Instance 1

Run this section of the script at Managed Instance 1 only after section 1 has completed successfully. It requires that the databases have already been created on both instances.

Note the command `exec sp_addsubscription ... @sync_type = N'automatic' ... @loopback_detection = 'true'` and that the `@sync_type` parameter is set to `automatic`. This means that the target subscriber database will be initialized with a snapshot. Also note how the `@loopback_detection` is enabled, which is needed for avoiding the re-replication of changes back to the originator.`

This section of the script includes the following steps:

- creates a sample table with test data
- creates the initial publication and configures a snapshot agent
- adds a subscription into the second Managed Instance and configures the Distribution Agent
- increases the login timeouts of the replication agents to 150 seconds (the default of 15 seconds is too short)
- starts the Snapshot and Distribution agents

```

/*****
/**** SECTION 2 - run this script at Managed Instance 1 ****
/****
-- Create the sample table and add a few rows
USE [replicationdb]
GO
CREATE TABLE ReplTest (
    ID INT NOT NULL PRIMARY KEY,
    c1 VARCHAR(100) NOT NULL,
    dt1 DATETIME DEFAULT (GETDATE())
)
GO
INSERT INTO ReplTest (ID, c1) VALUES (1, 'original data')
INSERT INTO ReplTest (ID, c1) VALUES (2, 'original data')
INSERT INTO ReplTest (ID, c1) VALUES (3, 'original data')
INSERT INTO ReplTest (ID, c1) VALUES (4, 'original data')
INSERT INTO ReplTest (ID, c1) VALUES (5, 'original data')
GO
SELECT * from ReplTest
GO

-- Add Publication
use [replicationdb]
exec sp_addpublication @publication = N'Publication_1', @description = N'Transactional publication on MI 1',
    @sync_method = N'concurrent', @retention = 0,
    @allow_push = N'true', @allow_pull = N'true', @allow_anonymous = N'true',
    @snapshot_in_defaultfolder = N'true', @compress_snapshot = N'false',
    @allow_subscription_copy = N'false',
    @repl_freq = N'continuous', @status = N'active', @independent_agent = N'true',
    @immediate_sync = N'true',
    @allow_sync_tran = N'false', @autogen_sync_procs = N'false', @allow_queued_tran = N'false', @allow_dts =
    @replicate_ddl = 1,
    @allow_initialize_from_backup = N'false', @enabled_for_p2p = N'false', @enabled_for_het_sub = N'false'
GO

-- Configure the Snapshot agent
exec sp_addpublication_snapshot @publication = N'Publication_1',
    @publisher_security_mode = 0,
    @publisher_login = 'TRANREPLADMIN', @publisher_password = '$strongPa11word',
    @job_login = 'TRANREPLADMIN', @job_password = '$strongPa11word',
    -- runs Sundays at 02:00
    @frequency_type = 8, @frequency_interval = 1, @frequency_relative_interval = 1, @frequency_recurrence_fac
    @active_start_time_of_day = 20000, @active_end_time_of_day = 235959,
    @active_start_date = 0, @active_end_date = 0
GO

-- Grant permissions to publication
exec sp_grant_publication_access @publication = N'Publication_1', @login = N'sa'
GO
exec sp_grant_publication_access @publication = N'Publication_1', @login = N'TRANREPLADMIN'
GO

-- Add the articles
use [replicationdb]
exec sp_addarticle @publication = N'Publication_1',
    @article = N'ReplTest', @source_owner = N'dbo', @source_object = N'ReplTest',
    @destination_table = N'ReplTest', @destination_owner = N'dbo',
    @type = N'logbased', @description = N'', @creation_script = N'',
    @pre_creation_cmd = N'drop',
    @schema_option = 0x00000000803509F, @identityrangemanagementoption = N'none',
    @status = 24, @vertical_partition = N'false',
    @ins_cmd = N'CALL [dbo].[sp_MSins_dboReplTest]', @del_cmd = N'CALL [dbo].[sp_MSdel_dboReplTest]', @upd_cm
GO

-- Add the subscription
-- This is a Push subscription into another Managed Instance
use [replicationdb]

```

```

exec sp_addsubscription @publication = N'Publication_1',
    @subscriber = N'secondmi', -- set the FQDN of the 2nd MI here
    @destination_db = N'replicationdb',
    @subscription_type = N'Push', @sync_type = N'automatic', @article = N'all',
    @update_mode = N'read only', @subscriber_type = 0
    , @loopback_detection = 'true' -- needed for bidirectional replication

-- Configure the Distribution Agent
exec sp_addpushsubscription_agent @publication = N'Publication_1',
    @subscriber = N'secondmi', -- set the FQDN of the 2nd MI here
    @subscriber_security_mode = 0,
    @subscriber_login = 'TRANREPLADMIN', @subscriber_password = '$trongPa11word',
    @job_login = 'TRANREPLADMIN', @job_password = '$trongPa11word',
    -- runs every 10 minutes
    @frequency_type = 4, @frequency_interval = 1, @frequency_relative_interval = 1, @frequency_recurrence_fac
    @frequency_subday = 4, @frequency_subday_interval = 10,
    @active_start_time_of_day = 0, @active_end_time_of_day = 235959,
    @active_start_date = 0, @active_end_date = 0,
    @enabled_for_syncmgr = N'False'

GO

-- apply increased login timeout setting to replication jobs
update msdb..sysjobsteps
    set command = command + N' -LoginTimeout 150'
    where subsystem in ('Distribution', 'LogReader', 'Snapshot')
    and command not like '%-LoginTimeout %'

GO

-- Start replication agents (if not already running)
USE [master]
GO
-- Snapshot agent
exec [replicationdb]..sp_startpublication_snapshot @publication = N'Publication_1'
GO
-- Distribution agent
exec [replicationdb]..sp_startpushsubscription_agent @publication = N'Publication_1',
    @subscriber = N'secondmi', -- set the FQDN of the 2nd MI here
    @subscriber_db = N'replicationdb'
GO

-- Check Replication Monitor here
-- confirm that the subscriber database at the second MI is initialized

/** END OF SECTION 2 *****/

```

Section 3 - Create the publication on Managed Instance 2 that loops back into the initial publication

Run this section of the script at Managed Instance 2, and run it only after Section 2 has completed successfully. It requires that the snapshot from the initial publication has been applied and the data table exists as created by the snapshot.

Note the command `exec sp_addsubscription ... @sync_type = N'replication support only' ... @loopback_detection = 'true'` and that the `@sync_type` parameter is set to replication support only. This means that the target subscriber database (= the initial Publisher database) already has the schema and data. Also note how the "`@loopback_detection`" is enabled, which is needed for avoiding the re-replication of changes back to the originator.

This section of the script includes the following steps:

- adds the secondary publication and publishes the table that has been created by the subscription from Managed Instance 1
- adds a subscription from the secondary publication back into the initial Publisher
- increases the login timeouts of the replication agents to 150 seconds (the default of 15 seconds is too short)
- starts the Distribution Agent job to close the replication cycle

```

/*****
/**** SECTION 3 - run this script at Managed Instance 2 *****/
/**** run this only after Section 2 has completed successfully ****/
/*****/

-- Add Publication
use [replicationdb]
exec sp_addpublication @publication = N'Publication_2', @description = N'Transactional publication on MI 2',
    @sync_method = N'concurrent', @retention = 0,
    @allow_push = N'true', @allow_pull = N'true', @allow_anonymous = N'true',
    @snapshot_in_defaultfolder = N'true', @compress_snapshot = N'false',
    @allow_subscription_copy = N'false',
    @repl_freq = N'continuous', @status = N'active', @independent_agent = N'true',
    @immediate_sync = N'true',
    @allow_sync_tran = N'false', @autogen_sync_procs = N'false', @allow_queued_tran = N'false', @allow_dts =
    @replicate_ddl = 1,
    @allow_initialize_from_backup = N'false', @enabled_for_p2p = N'false', @enabled_for_het_sub = N'false'
GO

-- configure the Snapshot agent
-- (does not need the Snapshot Agent, master data version is maintained on MI 1)

-- Grant permissions to publication
exec sp_grant_publication_access @publication = N'Publication_2', @login = N'sa'
GO
exec sp_grant_publication_access @publication = N'Publication_2', @login = N'TRANREPLADMIN'
GO

-- Add the articles
use [replicationdb]
exec sp_addarticle @publication = N'Publication_2',
    @article = N'ReplTest', @source_owner = N'dbo', @source_object = N'ReplTest',
    @destination_table = N'ReplTest', @destination_owner = N'dbo',
    @type = N'logbased', @description = N'', @creation_script = N'',
    @pre_creation_cmd = N'drop',
    @schema_option = 0x000000000803509F, @identityrangemanagementoption = N'none',
    @status = 24, @vertical_partition = N'false',
    @ins_cmd = N'CALL [dbo].[sp_MSins_dboReplTest]', @del_cmd = N'CALL [dbo].[sp_MSdel_dboReplTest]', @upd_cm
GO

-- Add the subscription
-- This is a Push subscription into another Managed Instance
use [replicationdb]
exec sp_addsubscription @publication = N'Publication_2',
    @subscriber = N'firstmi', -- set the FQDN of the first MI here
    @destination_db = N'replicationdb',
    @subscription_type = N'Push', @sync_type = N'replication support only', @article = N'all',
    @update_mode = N'read only', @subscriber_type = 0
    , @loopback_detection = 'true' -- needed for bidirectional replication

-- Configure the Distribution Agent
exec sp_addpushsubscription_agent @publication = N'Publication_2',
    @subscriber = N'firstmi', -- set the FQDN of the first MI here
    @subscriber_security_mode = 0,
    @subscriber_login = N'TRANREPLADMIN', @subscriber_password = N'$trongPa1lword',
    @job_login = N'TRANREPLADMIN', @job_password = N'$trongPa1lword',
    -- runs every 10 minutes
    @frequency_type = 4, @frequency_interval = 1, @frequency_relative_interval = 1, @frequency_recurrence_fac
    @frequency_subday = 4, @frequency_subday_interval = 10,
    @active_start_time_of_day = 0, @active_end_time_of_day = 235959,
    @active_start_date = 0, @active_end_date = 0,
    @enabled_for_syncmgr = N'False'
GO

-- apply increased login timeout setting to replication jobs
update msdb..sysjobsteps
    set command = command + N' -LoginTimeout 150'
where subsystem in ('Distribution', 'LogReader', 'Snapshot')

```



```
and command not like '%-LoginTimeout %'
GO

-- Start replication agents (if not already running)
USE [master]
GO

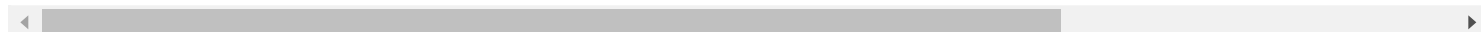
-- Snapshot agent
-- (not available/needed)
GO

-- Distribution agent
exec [replicationdb]..sp_startpushsubscription_agent @publication = N'Publication_2',
    @subscriber = N'firstmi', -- set the FQDN of the first MI here
    @subscriber_db = N'replicationdb'
GO






-- Check Replication Monitor here

/**** END OF SECTION 3 *****/

/*****/
/**** END OF SCRIPT *****/
/*****/
```



References

- [Bidirectional Transactional Replication](#) 
- [sp_addsubscription \(Transact-SQL\)](#)  and specifically its '[@sync_type](#)' parameter 
- [Transactional Replication](#) 
- [Best Practices for Replication Administration](#) 

How good have you found this content?



-