|  |
| --- |
| yogev cohen  ‏19/3/2019 |

|  |
| --- |
|  |
| **SQL Server Transactional Replication** | |
|  |

**1. All about SQL Server Replication:**

In this article we will review the features of Transaction Replication and how to get started with it. Replication was first published in 2008 and available only in standard edition, replication in SQL Server replicates or data duplicates. Whenever you need to make a copy of your data, or duplicate its modification, replication can be used. You can create this copy on the same database or on a separate server.The copy can be saved in a synchronized with the source data or synchronized at scheduled intervals. One-way synchronization, as well as two-way synchronization, is possible. Replication can also be used to keep multiple arrays synchronized with each other.

There are several scenarios where the use of SQL Server replication is a good choice. For example, you can use cloning to copy your data to another computer for viewing reporting or to provide a source for Analysis Services. Using replication in this way offers two main advantages. First, you separate your heavy reporting queries from your online transaction processing, maintaining intensive online processing and transaction load on one system and intense reporting load on the other.The other benefit is due to the fact that replication can be planned to run at certain intervals. For example, you can run a clone every night to provide a snapshot of data to the reporting environment. As a result, your reports will only contain data from completed days, and will not contain data from part of the day. This means that you do not have to build additional logic into your system to deal with today incomplete data.

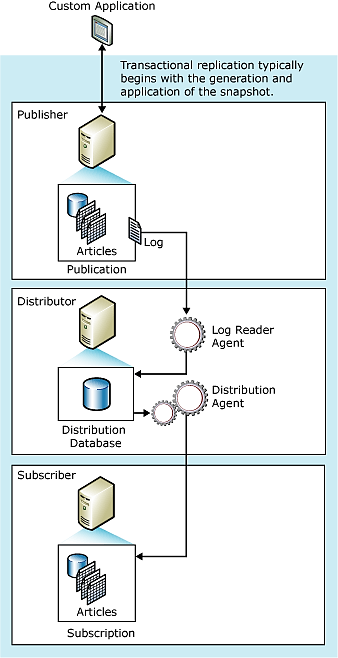
**1.1 Alternative technologies to replication:**

An alternative to replication, you can synchronize databases using the Microsoft Sync Framework. Sync Framework includes intuitive and flexible API components and APIs that enable easy synchronization between SQL Server and SQL Azure databases. The Sync Framework also includes options that can be customized to synchronize a SQL Server database with any other database that is compatible with ADO.NET.

Two additional replication features have been added in SQL Server 2008, tracking changes and changing data capture. These two technologies are completely separate from reproducing**:  
  
The tracking change**, just like transaction replication, is available in the Standard Edition. Tracking change is an easy solution that provides an effective tracking mechanism of change for applications that allow applications to query changes in database data and access to change-related information, application developers were required to implement custom tracking mechanisms for change.  
 **Change data** capture is designed to recognize insert, update, and delete activity on SQL Server tables and to make the details of the changes available in an easily consumed relational format. The change tables used by change data capture contain columns that mirror the column structure of a source table, along with the metadata needed to understand the changes that have occurred. Change data capture is available only on the Enterprise, Developer, and Evaluation editions of SQL Server.

Another feature is **Database mirroring**. Can be used it together with replication to improve availability for the publication database. Database mirroring involves two copies of a single database that defined on different computers. At any given time, only one copy of the database is currently available to users. This copy is known as the principal database. Updates made by users to the principal database are applied on the other copy of the database, known as the mirror database.

**1.2 Replication Types:**There are three main types of replication available in SQL Server:

1. snapshot replication - Snapshot replication creates a complete copy of the objects and their data each time it runs. It is writing the contents of each table into the shared snapshot folder. The snapshot folder is a shared folder location that must be set up on the distributor when enabling replication. Each participant in a replication setup needs to have access to the snapshot folder. Every time snapshot replication is running, everything is recopied from scratch, so it has high bandwidth and storage requirements. All other types of replication use by default a single replication snapshot to sync up all subscribers with the distributor only during the initial setup.
2. Merge replication - Merge replication was designed to allow for changes of the data to be made on the publisher as well as the subscriber. Merge replication also allows for disconnected scenarios, where a subscriber might not be connected during the day. That subscriber would synchronize after reconnecting. If a row gets updated in two different places at the same time, a conflict occurs. Merge replication comes with several built-in options to resolve those conflicts.
3. **Transactional Replication** - Transactional replication works on a transaction basis. Every committed transaction gets scanned for changes applied to replication articles. Scanning of the changes is done by a log reader agent, which reads through the transaction log of the publisher database. If there are changes on the published object, those changes get logged on the distributor in the distribution database. From there they make their way to the subscribers. Transactional replication allows for close to real time synchronization. While there are several options to allow for bidirectional data movement, transactional replication was originally designed to work one way only.   
   The process of Transactional Replication can be seen in the next figure.  
     
   

* Peer - to - peer replication and Oracle publishing replication are variations of these three types of replication.

**2. Working with SQL Server Transactional Replication:**

SQL server transactional replication used database as publication.  
Any selected object on publication is an article. It can be a table or any database objects. Transactional replication also needs a distributor that Serves as a messenger, keep track of delivery status and track history to synchronization. Article can be stored as files or in a folder which both publisher and subscribers have access, or as tables in a distribution database.

**2.1 Transactional replication components:**

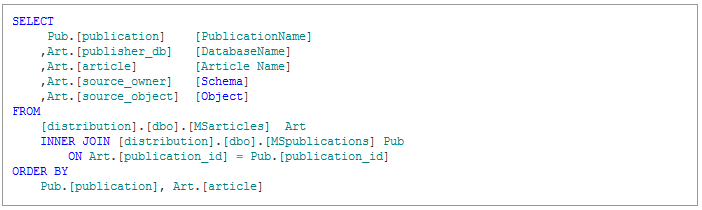
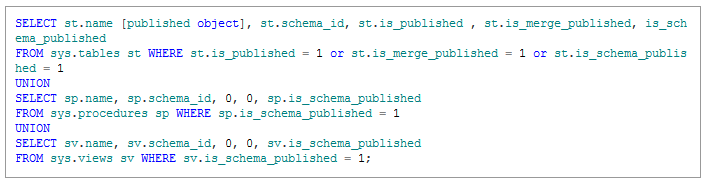
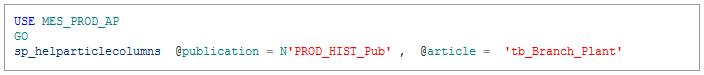
SQL server replication is combined of several key components that are grouped into the following topics:

* **Replication roles**
* **Replication data**
* **Replication agents**
* **Replication maintenance jobs**

**Replication roles:**There are three key roles in replication

1. Publisher - The Publisher is a database that makes data available to other locations. The Publisher can have one or more publications.
2. Distributor - The Distributor is a database that acts as a station for replication specific data associated with one or more Publishers. The distributor is a single database that acts as both the Publisher and the Distributor. Each Publisher is associated with a single database known as a distribution database.   
   **distribution database** - Each Distributor must have at least one distribution database. The distribution database consists of article detail, replication meta-data and data. A Distributor can hold more than one distribution database.  
     
     
     
     
     
     
   Here are some examples of system queries and setup:  
   server is a distributor or not:  
   distribution database installed or not:  
     
   
3. Subscriber - Subscriber is database instance that consumes replication data from a publication is called a Subscriber. The subscriber can receive data from one or more publishers and publications. The subscriber can also transfer data changes back to the publisher or republish the data to other subscribers depending on the type of the replication design and model.

**Replication data:**

1. Article **-** An article is the smallest set of data that can be configures for replication. It can be table, a view or stored procedure and can have additional restrictions on the rows and columns included in each article.  
   To list all the articles that are published, run the following T-SQL:  
     
     
     
     
     
     
     
     
   To get the details of articles in transactional replication in a published database, run the following T-SQL:  
     
     
     
     
     
     
     
     
     
     
     
   To get detailed information about an article in the listed publisher, run the following T-SQL:  
     
     
     
     
     
     
     
     
     
     
   To get column level details, run the following T-SQL:  
     
   
2. Publication **–** Publication is a grouping of articles published together. Using a publication enables the replication of logically grouped articles to be managed rather than having to manage each article individually.
3. Subscription **–** Subscription is a request to receive data from one or more publications. It can add additional constrain to the publication.  
   There are two types of subscriptions: push subscriptions and pull subscriptions

* Push subscription - Distributor directly updates the data in the Subscriber database
* Pull subscription - The Subscriber is scheduled to check at the Distributor if any new changes are available, and then updates the data in the subscription database.

## **Replication agents:**

SQL Server replication agents run as scheduled jobs under SQL Server Agent. Replication agents can also be running from the command line and by applications that use Replication Management Objects (RMO). SQL Server replication agents can be monitored using Replication Monitor and SQL Server Management Studio.

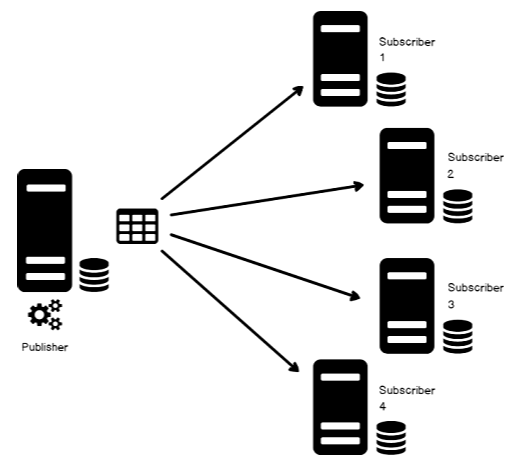
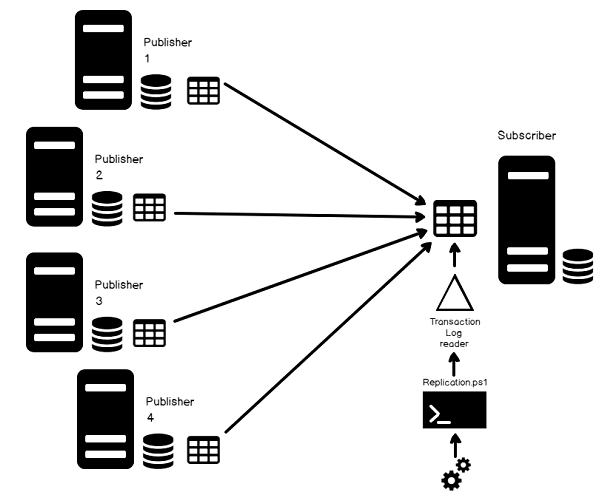
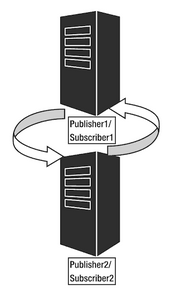
1. Snapshot agent - The Replication Snapshot Agent is used with all types of SQL Server replication technology as it provides the required data set to perform the initial data synchronization of the publication database with the subscription database. It prepares schema and initial data of published articles, snapshot files, and records information about the synchronization type in the distribution database.
2. Log reader agent - The Log Reader Agent is used only with transactional replication. It moves replication transactions from the online transaction log of the publication database to the distribution database.
3. Distribution agent – This agent is executed by SQL Agent job that reads the transactions written to the distribution database and applies them to the subscribing databases in transactional replication.
4. Queue reader agent - This agent is used to read messages stored in SQL Server queue or a Microsoft Message Queue. It then applies those messages to the publisher. Queue reader agent is used in snapshot or transactional replication publications.

**Replication maintenance jobs:**

We have a number of jobs that help us perform maintenance in transactional replication mode:

1. Agent history cleanup – This job removes replication agent history that is stored in the distribution database.
2. Distribution cleanup – This job removes transactions from the distribution database after they are no longer needed.
3. Expired subscription cleanup – This job determines when a snapshot has expired and removes it.
4. Reinitialize failed subscription – This job looks for subscriptions that have failed and marks them for initialization.
5. Replication agent monitor – This job monitors the execution of the SQL agents and writes to the Windows event log when a job fails.
6. Replication agent cleanup – This job monitors the execution of the replication agents on a distributor.

**2.2 Replication models:**

1. Central Publisher and Distributor Model - In the central Publisher and Distributor model (the SQL Server default model), one or two servers are defined as the Publisher and Distributor. The Publisher and Distributor publishes and distributes data to one server or number of servers that are set up as Subscribers.  
   The Publisher and the Distributor can be defined on one server or on separate servers. In either case, the publication server is the primary owner or source of all replicated data. Data that is received at subscription server is intended to be read only.
2. Central Subscriber and Multiple Publishers Model - In the central Subscriber and multiple Publishers model, multiple Publishers replicate data to a single Subscriber. Because multiple Publishers are writing to the same subscription table, it is important to ensure that all data has a unique local owner so that another Publisher does not overwrite it.
3. Multiple Publishers and Multiple Subscribers Model - In this model, multiple publication servers and multiple subscription servers each potentially play a role. This model is the closest implementation to fully distributed data processing. The design of both the schema and update types need to sure that an appropriate level of data consistency exists at all servers.

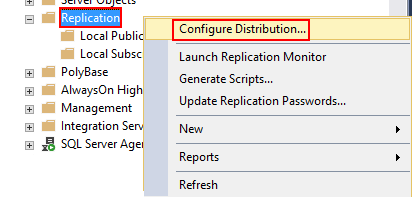
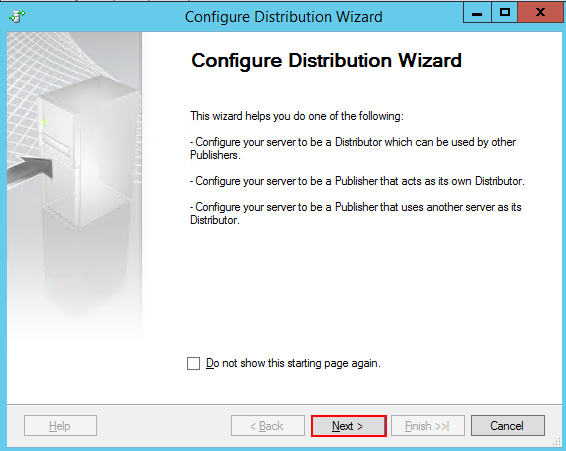
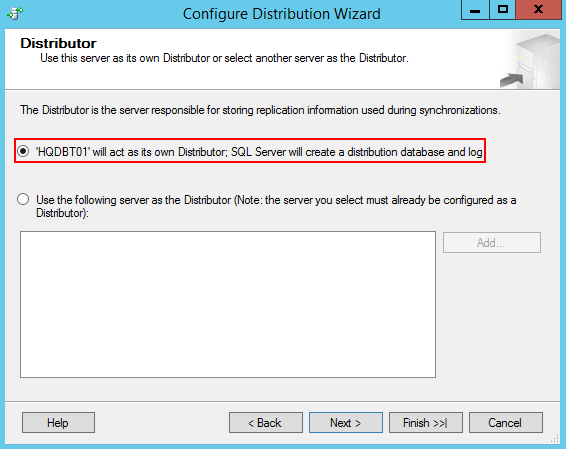
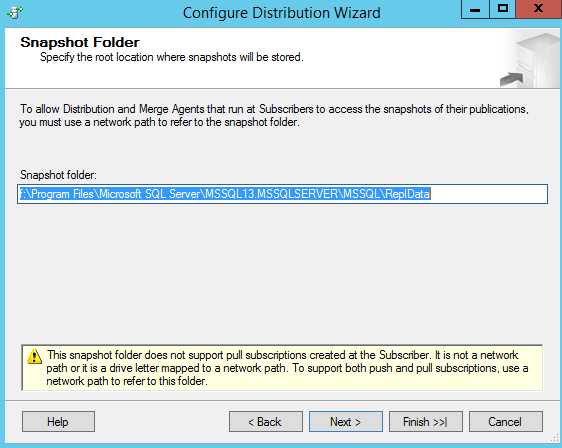
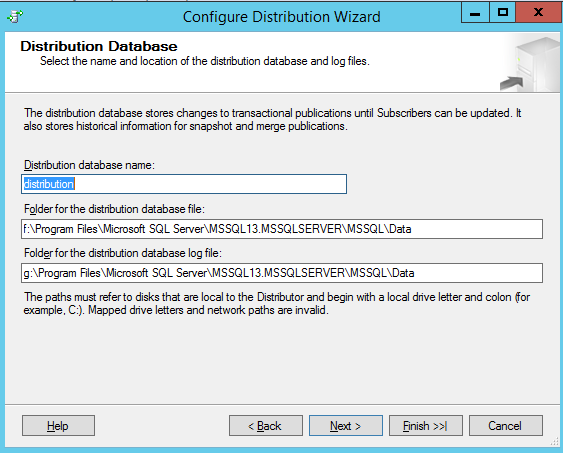
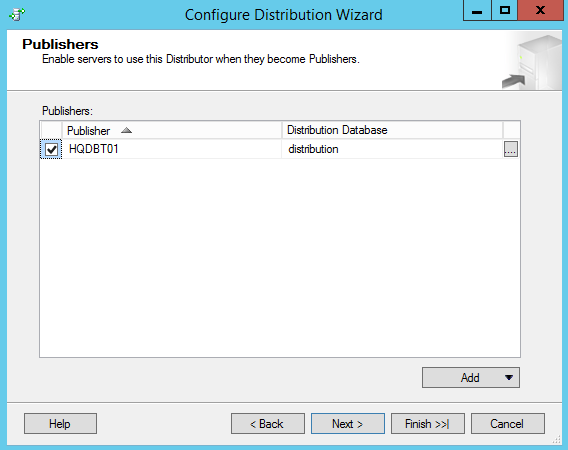
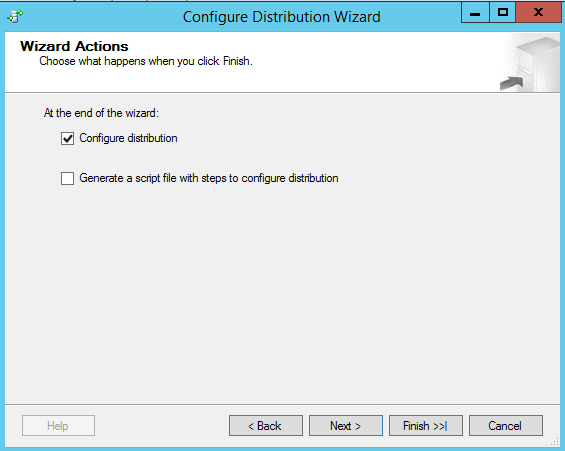
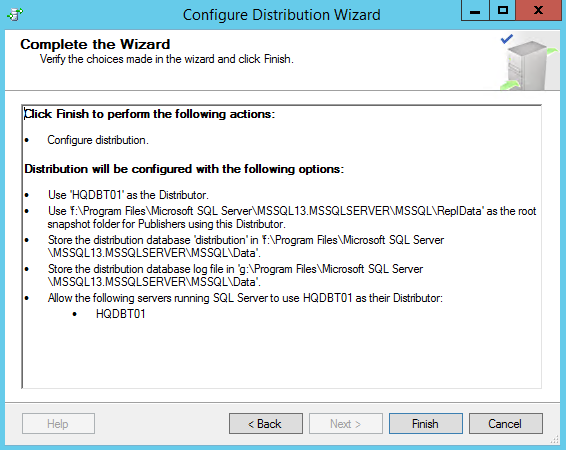
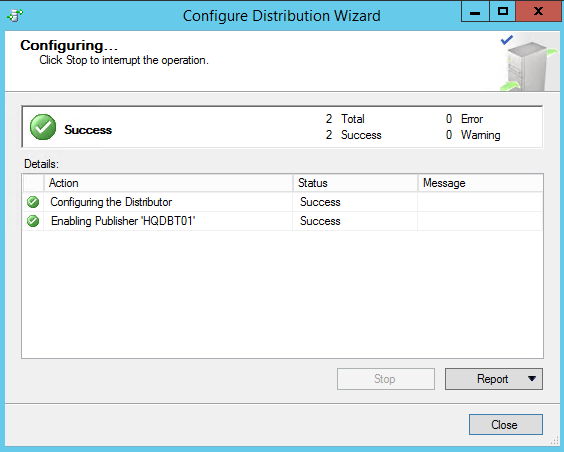
## **2.3 Pre-requisites**

1. The account must at minimum be a member of the database owner fixed database role in the SQL replication Publisher, Distributor and Subscriber databases
2. For securing the replication snapshot folder using a Snapshot Agent, the account must have read and write or modify permission on the replication snapshot shared folder.
3. At least one database should have an article and must have a Primary Key, a basic principle that every article should have a Primary Key is considered as best candidate for Transactional Replication. The primary key is used to maintain uniqueness of records.
4. Scheduling the agent and jobs
5. Enough disk space for the databases being published, It is required to make sure that we have enough space available for the SQL transaction log for the published database.

# 3. setup and configuration Transactional Replication:

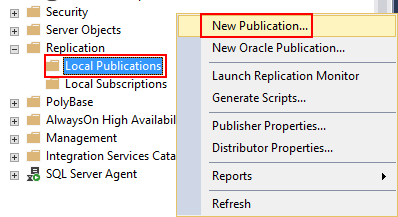
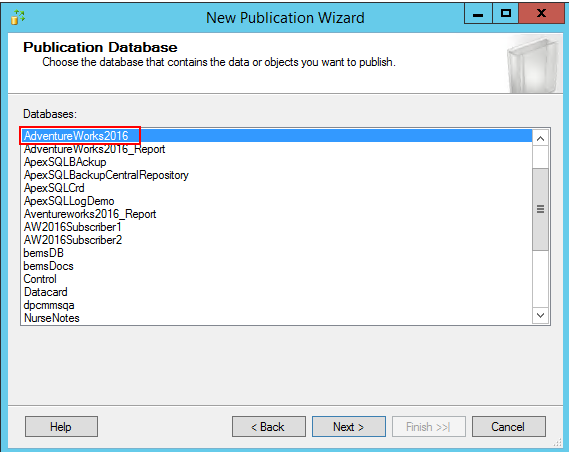
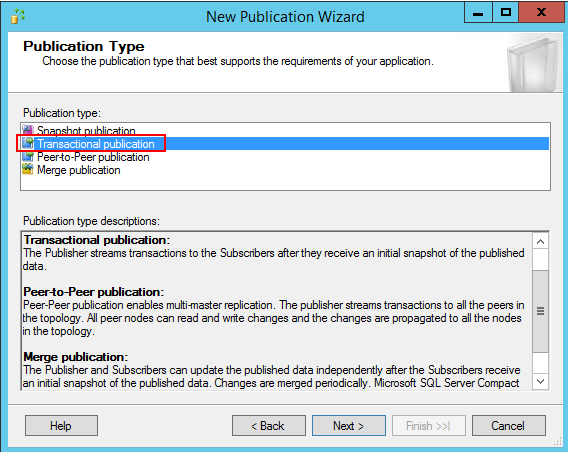
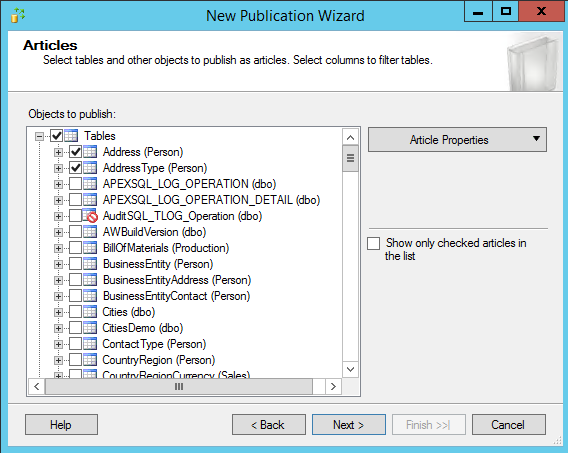
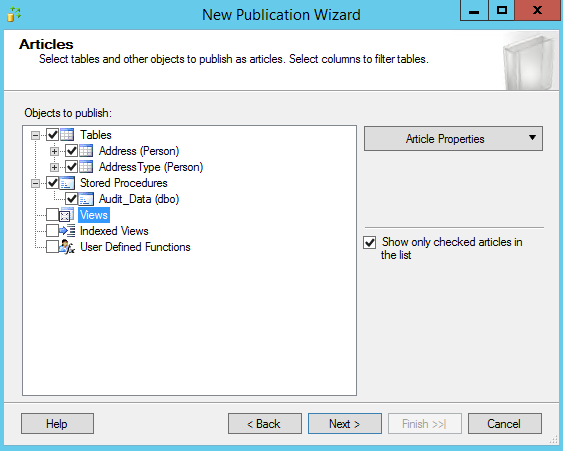
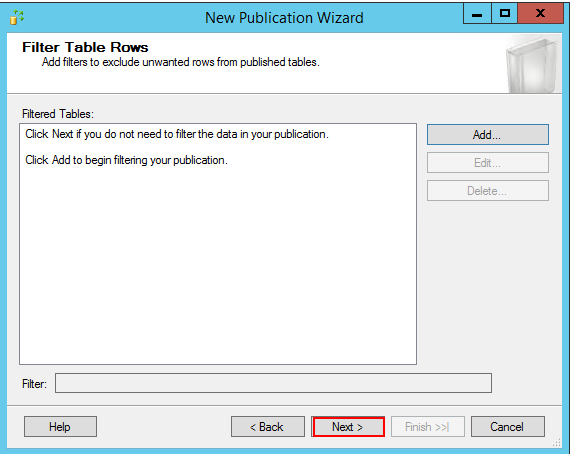
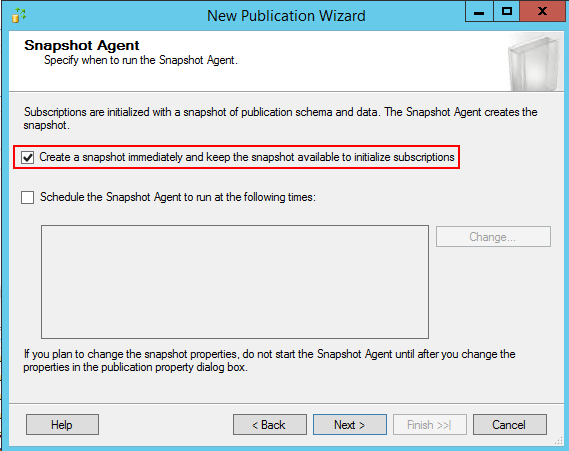
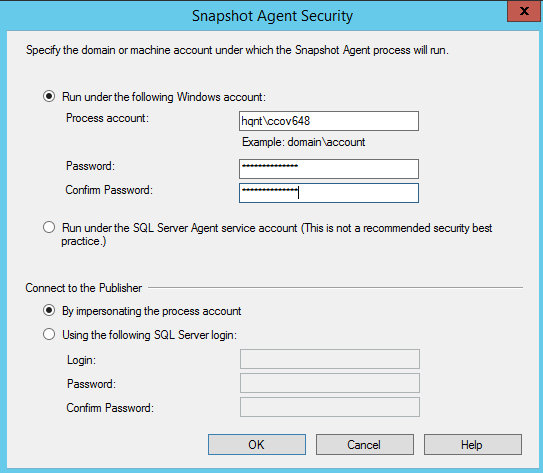
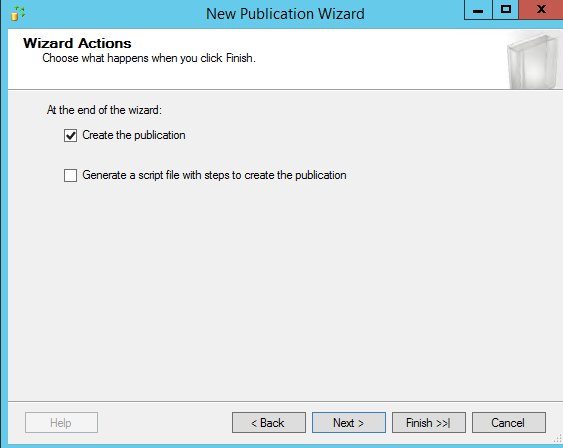
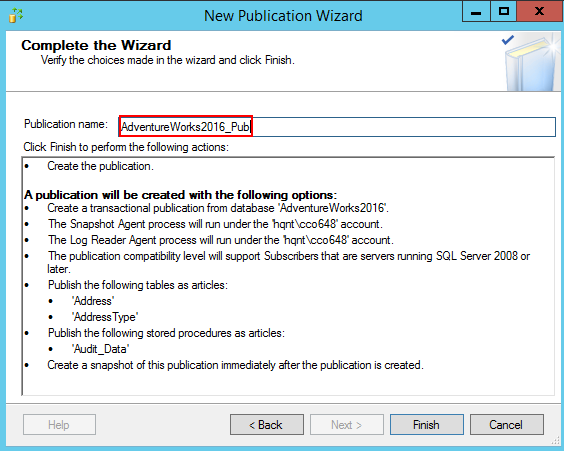
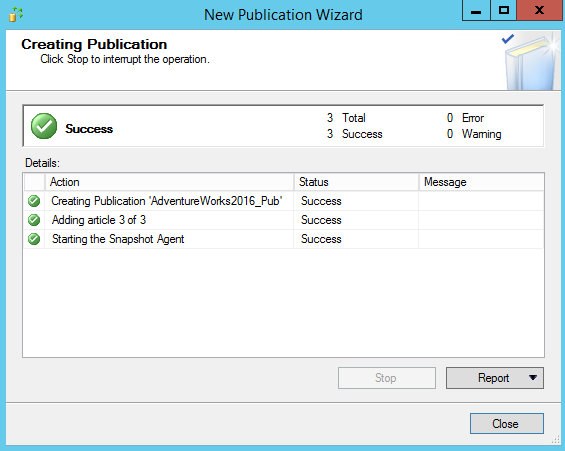
To first set up transaction replication, required to configure the replication Distributor and create a replication Publication, replication snapshot sherd folder and a replication Subscription, the following steps show how to set all of the above.

## **3.1 Configure Distributor:**

1. Open SQL Server Management Studio (SSMS) and connect to the SQL Server instance
2. In **Object Explorer**, browse to the replication folder, right-click the **Replication** folder and click ***Configure Distribution***
3. The first page is the **Distribution Configuration Wizard**. The pages outline the general details about configuration distributor.
4. On the **"Distributor**" page, there is an option to choose to set up the current instance to be a Distributor or select another instance that is already been configured as a Distributor. It is necessary to select the desired option and then Click ***Next***.
5. in the configure **"Snapshot Folder"**, type in the path of the replication snapshot folder or leave the default path of the SQL replication snapshot folder. The snapshot folder is used for initial data synchronization of transactional replication and make sure it is large enough to hold all the replicated data. Click ***Next***.
6. Now, configure the replication distribution database. Specify the name of the distribution database and the folders where the data and log files should be located. No change in the default values is required. Click ***Next***.
7. In the **"Publishers**" page, specify the replication Publishers that are going to access the Distributor and Click ***Next***.
8. In **"Wizard Actions"**, there is an option to run immediately or create a script that can be executed at a later time and Click ***Next*.**
9. In the "**complete the wizard**" page, review the settings and configuration options and then click ***Finish*** to enable the Distributor.
10.  The following page appears and it shows the progress of the distributor setup.

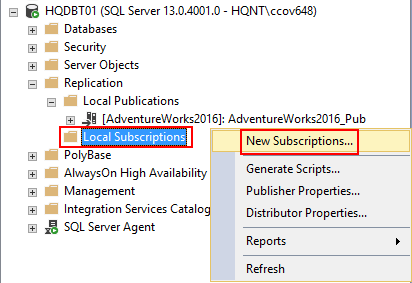
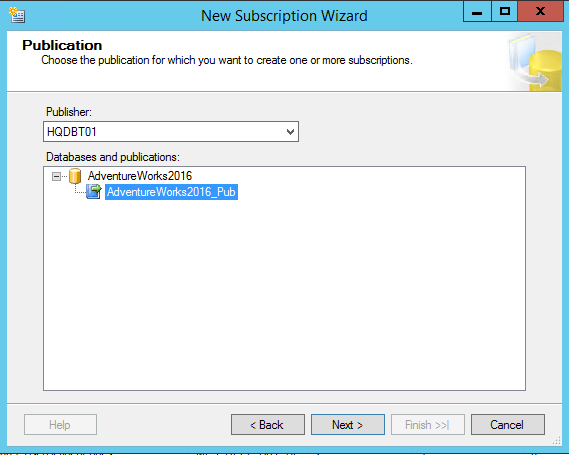
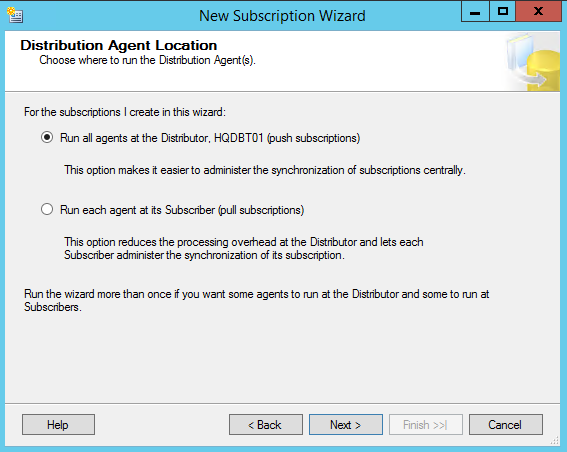
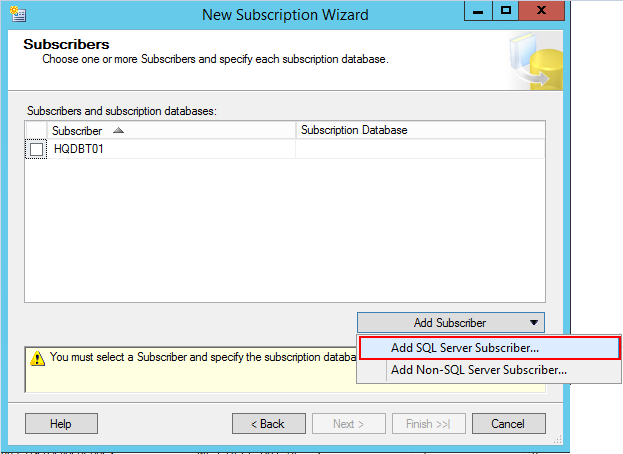
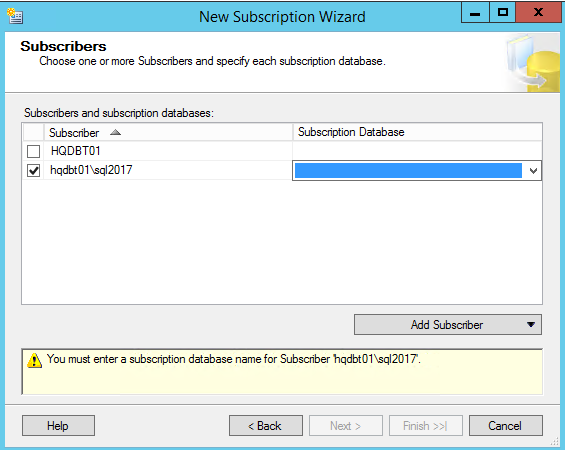
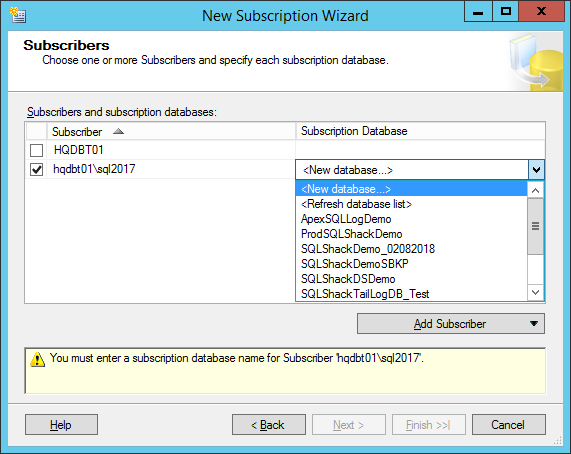
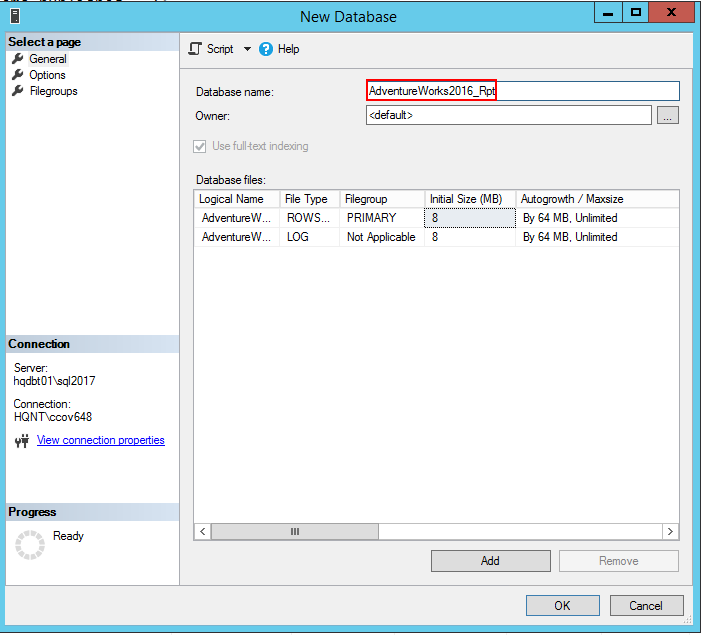
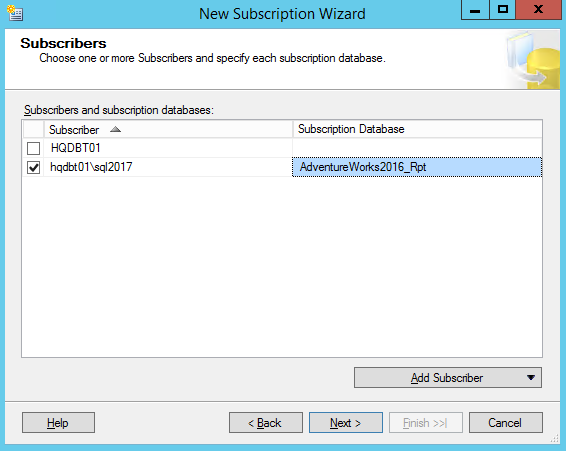
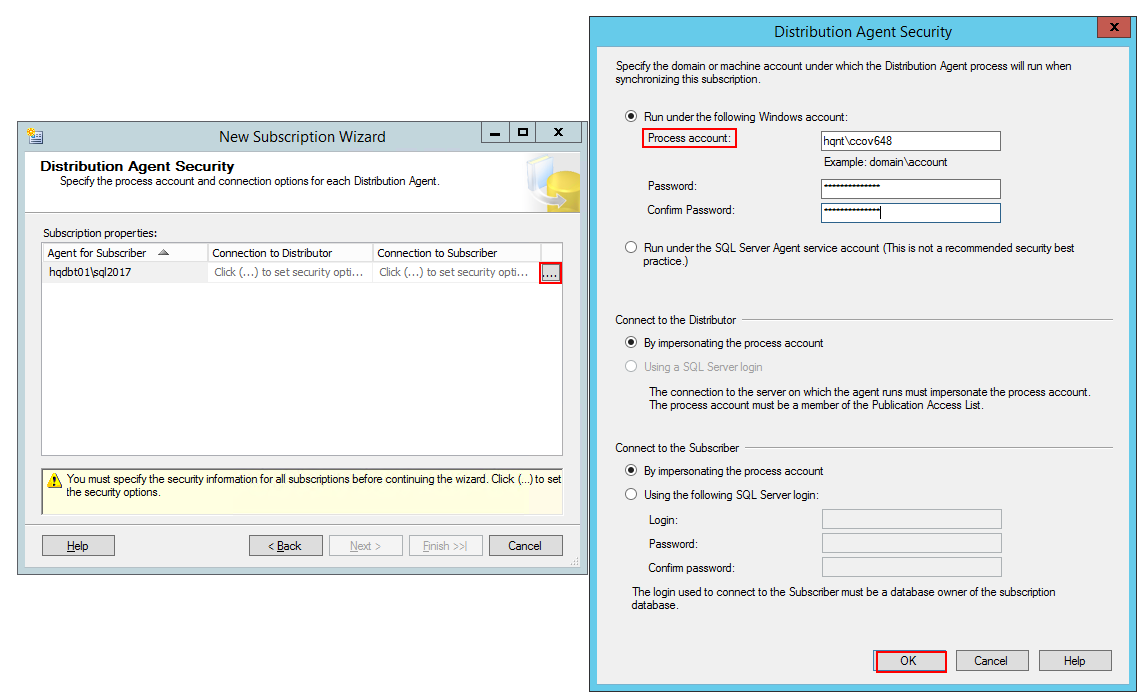
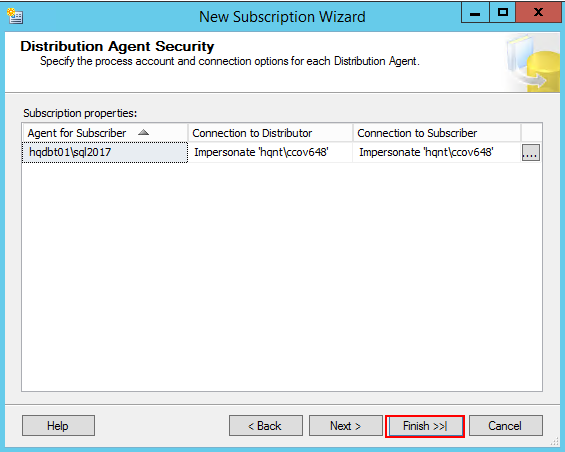
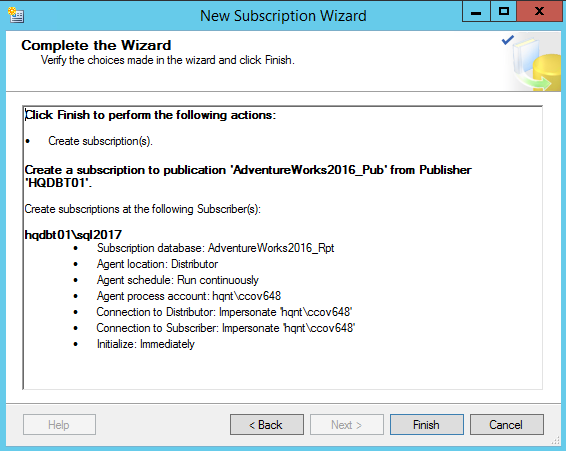
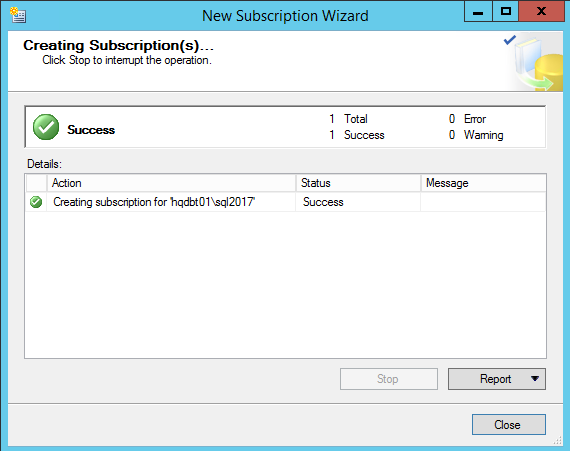
## **3.2 Configure Distributor:**

Once the Distributor is configured, we need to create a publication following these steps:

1. In **"Object Explorer"**, locate the **Replication folder**, right-click **Local Publication**, and then click ***New Publication***.
2. Next, the "**New Publication Wizard"** appears and outlines the general information about creating Publication.
3. In the "**Publication Database"** page, select the database and click ***Next***.
4. On the **"Publication Type**" page, select ***Transactional publication***, and click ***Next***.
5. Now, on the **"Articles**" page, choose the articles should be part of this publication, and click ***Next***.
6.  After the objects selection, Check the **"Show only checked articles in the list"** optionto list candidates of Publication, and click ***Next***.
7. Next, in the **"Filter Table Rows** pages", define filters that should be applied to your articles. and click ***Next***.
8. In the **"Snapshot Agent"** page specify when to run the Snapshot Agent. It can be run immediately or it can schedule to run at a later time. In this case, and click ***Next***.
9. Now, in the **"Agent Security**" page, specify the account to use and run the Snapshot Agent using **"Security setting",** and click ***OK***.
10. In the complete **"Wizard Actions**" page, there are two options. You can create the Publication immediately or save the configuration in the script file to run at a later time, and click ***Next***.
11. In the final page type in the publication name and Click ***Finish***.
12. In the "**Creating Publication**" page, there is information about the wizard’s progress as it works through each step of the process.

## **3.3 Configure Subscriber:**

The final step in setting up transaction replication is to create the subscription:

1. In Object Explorer, expand the Replication folder, right-click Local Subscriptions, and then click ***New Subscriptions***.
2. The "New Subscription Wizard" appears outlines the general information about the wizard, click ***Next***.
3. On the **"Publication**" page, select Publication and then click ***Next***.
4. On the "Distribution Agent location" page, select “***Run all agents at the Distributor*”** and Click ***Next***.
5. On the **"Subscribers**" page, select ***Add Subscriber*** and then select ***Add SQL Server Subscriber*** from the **Add Subscriber button**. This step opens the Connect to Server dialog box. Enter the subscriber instance name and then select Connect.  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   
6. After the subscriber instance has been added, select the drop-down next to the instance name of your subscriber. Then select New Database under Subscription Database and type in the database name and then Click ***Ok***.
7. The subscription database is created and registered to the subscriber. Now, Click ***Next***. Make sure that the account has database owner permission on the newly created database.
8. On the "**Distribution Agent Security**" page, select the ellipsis button. Type in the process account details and Click ***Ok***.
9. Select ***Finish*** accepting the default values on the remaining pages and completing the wizard.
10. On the "complete the wizard" page, we can see the detailed summary of the newly created subscription. Click ***Finish***.
11. In the **"Creating Subscription(s)"** page should show that the process has been successful or not.

We have successfully configured Publisher, Local Distributor and a remote Subscriber **transactional replication model**. You can now insert, delete, or update the data in the Address and Address Type articles to see the data propagation mechanism between the servers.

## **4. Summary**

This article review the fundamentals of configuring transactional replication through an understanding of concurrent snapshot processing, the internal of transaction log and the different agents involved in processing transactions from the publisher server by the distributing server to the subscribing servers.