Author: Rafael Oliveira (raolivei)

Replicating Process Orchestrator databases with SQL Server Replication

# DESCRIPTION

This allow OrchestratorProcess and OrchestratorReporting databases to be replicated to other(s) SQL Server nodes using SQL Server Replication feature.

## SQL Server 2005 Features

Restartable Snapshots

Oracle Publishing

Replicating all DDLs

Merge Replication allows to introduce custom business logic into the synchronization process

Merge Replication provides the ability to replicate data over HTTP with web synchronization option

Updatable Transactional Subscriptions can now handle updates to large data types at Subscribers

## SQL Server 2008 Features

In SQL Server 2005, replication had to be stopped in order to perform some actions like adding nodes, making schema changes, etc. But in 2008, these can be done online.

Conflict detection capacity in peer-to-peer replication.

All types of conflicts are detected and reported through agent error log or conflicts table.

In SQL Server 2005, switch partition is unsupported, but in 2008 it supports.

@allow\_partition\_switch

@replicate\_partition\_switch

Performance improvements, under Windows 2008

Snapshot delivery of more than 500MB/minute

Time to deliver 100000 varbinary(max) records in less than 2minutes where in 2005 223 minutes.

## SQL Server 2012 Features

Updatable subscriptions with transactional publications are discontinued.

Four new stored procedures provide replication support for AlwaysOn.

sp\_get\_redirected\_publisher

sp\_redirect\_publisher

sp\_validate\_replica\_hosts\_as\_publishers

sp\_validate\_redirected\_publisher

Replication supports the following features on Availability groups:

A publication database can be part of an availability group. The publisher instances must share a common distributor.

In an AlwaysOn Availability Group, an AlwaysOn secondary cannot be a publisher. Republishing is not supported when replication is combined with AlwaysOn.

Heterogeneous replication to non-SQL Server subscribers is deprecated. To move data, create solutions using change data capture and SSIS.

Oracle Publishing is deprecated.

# REPLICATION ENTITIES

SQL Server replication is based on the “Publish and Subscribe” mechanism.

## Publisher

It is a source database where replication starts. It makes data available for replication.

Publishers define what they publish through a publication.

## Publication

Group of database objects included in replication like tables, views, indexes, etc.

## Distributor

It is intermediary between publisher and subscriber.

It receives published transactions or snapshots and then stores and forwards these publications to the subscriber.

## Subscriber

It is the destination database where replication ends.

It can subscribe to multiple publications from multiple publishers.

It can send data back to publisher or publish data to other subscribers.

## Subscription

It is a request by a subscriber to receive a publication.

We have two types of subscriptions - push and pull.

### Push Subscriptions

With this subscription, the publisher is responsible for updating all the changes to the subscriber without the subscriber asking those changes.

Push subscriptions are created at the Publisher server

### Pull Subscriptions

With this subscription the subscriber initiates the replication instead of the publisher.

The subscriptions are created at the Subscriber server.

# REPLICATION AGENTS

## Snapshot Agent

* It is an executable file that prepares snapshot files containing schema and data of published tables and db objects.
* It stores the files in the snapshot folder, and records synchronization jobs in the distribution database.

## Distribution Agent

* It is used with snapshot and transactional replication.
* It applies the initial snapshot to the Subscriber and moves transactions held in the Distribution db to Subscribers.
* It runs at either the Distributor for push subscriptions or at the Subscriber for pull subscriptions.

## Log Reader Agent

* It is used with transactional replication, which moves transactions marked for replication from the transaction log on the publisher to the distribution db.
* Each db has its own Log Reader Agent that runs on the Distributor and connects to the Publisher.

## Merge Agent

* It is used with merge replication.
* It applies the initial snapshot to the Subscriber and moves incremental data changes that occur.
* Each merge subscription has its own Merge Agent that connects to both the Publisher and the Subscriber and updates both.
* It captures changes using triggers.

## Queue Reader Agent

* It is used with transactional replication with the queued updating option.
* It runs at the Distributor and moves changes made at the Subscriber back to the Publisher.

Unlike Distribution Agent and Merge Agent, only one instance of the Queue Reader Agent exists to service all Publishers and publications for a given distribution db.

# DEFINE REPLICATION TYPE

* Snapshot publication:

The Publisher sends a snapshot of the published data to Subscribers at scheduled intervals.

* Transactional publication:

The Publisher streams transactions to the Subscribers after they receive an initial snapshot of the published data.

* Peer-to-Peer publication:

Peer-Peer publication enables multi-master replication. The publisher streams transactions to all the peers in the topology. All peer nodes can read and write changes and the changes are propagated to all the nodes in the topology.

* Merge publication:

The Publisher and Subscribers can update the published data independently after the Subscribers receive an initial snapshot of the published data. Changes are merged periodically. Microsoft SQL Server Compact Edition can only subscribe to merge publications.

# ARCHITECTURE

## Snapshot Replication Architecture



# Environment

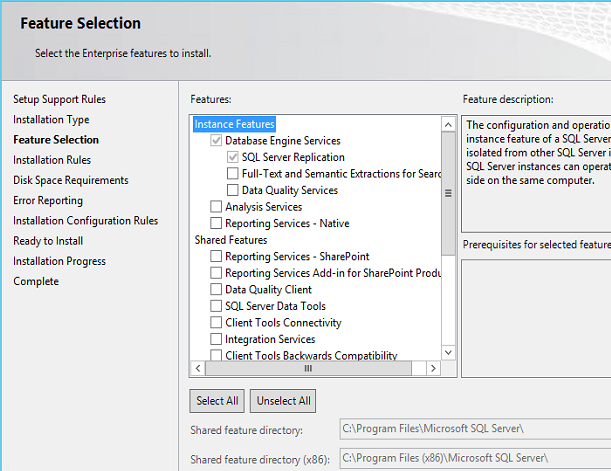
Distribution will be configured with the following options:

* Use ‘VEGA-1’ as Publisher;
* Use 'VEGA-2' as Distributor;
* Use ‘VEGA-2’ as Subscriber;
* Configure the SQL Server Agent service on 'VEGA-2' to start automatically when the computer is started;
* Use **Snapshot Publication** replication type;
* Use '[\\vega-2\ReplData](file:///\\vega-2\ReplData)' as the root snapshot folder for Publishers using this Distributor;
* Store the distribution database 'distribution' in 'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Data';
* Store the distribution database log file in 'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Data';
* Allow the following servers running SQL Server to use VEGA-2 as their Distributor: VEGA-1 (publisher);

# Prerequisites

## SQL Server Replication

Make sure all prerequisites are passed and install ‘SQL Server Replication’ feature



## Passive PO installation

Install the passive PO system – use the same PO database names you used in your active PO system (e.g: OrchestratorProcess and OrchestratorReporting)

# Configuring Replication

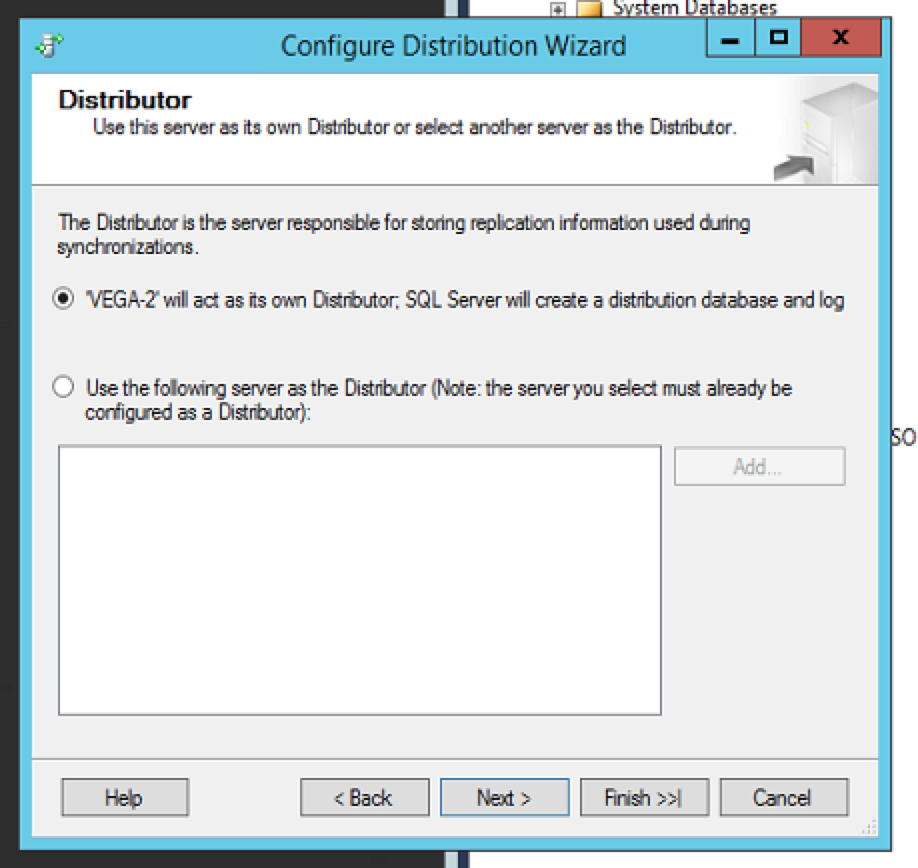
* Configuring distributor
* Configuring publisher
* Creating publication of required type
* Creating subscription

## Step 1: Configuring distributor and publisher

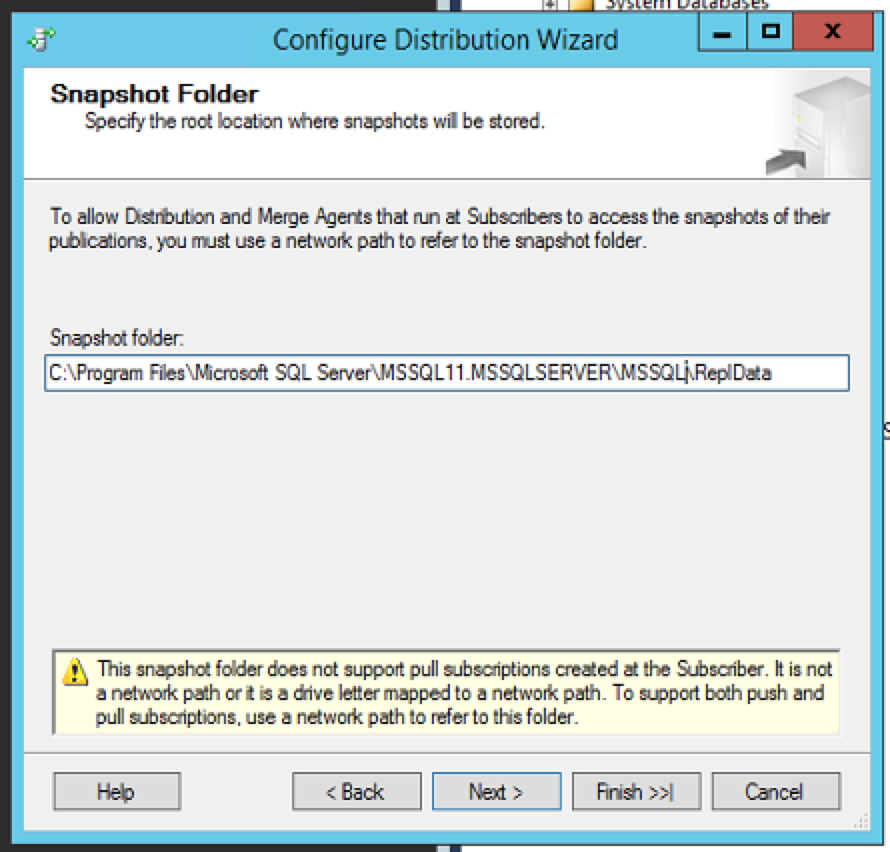
1. Using Microsoft SQL Server Management Studio On VEGA-1, connect to VEGA-2;
2. Right click on Replication -> Configure Distribution…;

## /var/folders/m9/wp8hkpqn5ygcj8n92g11htc80000gp/T/ro.nextwave.Snappy/ro.nextwave.Snappy/Windows 10 x64 VMware Fusion, Today at 12.58.00 AM.png

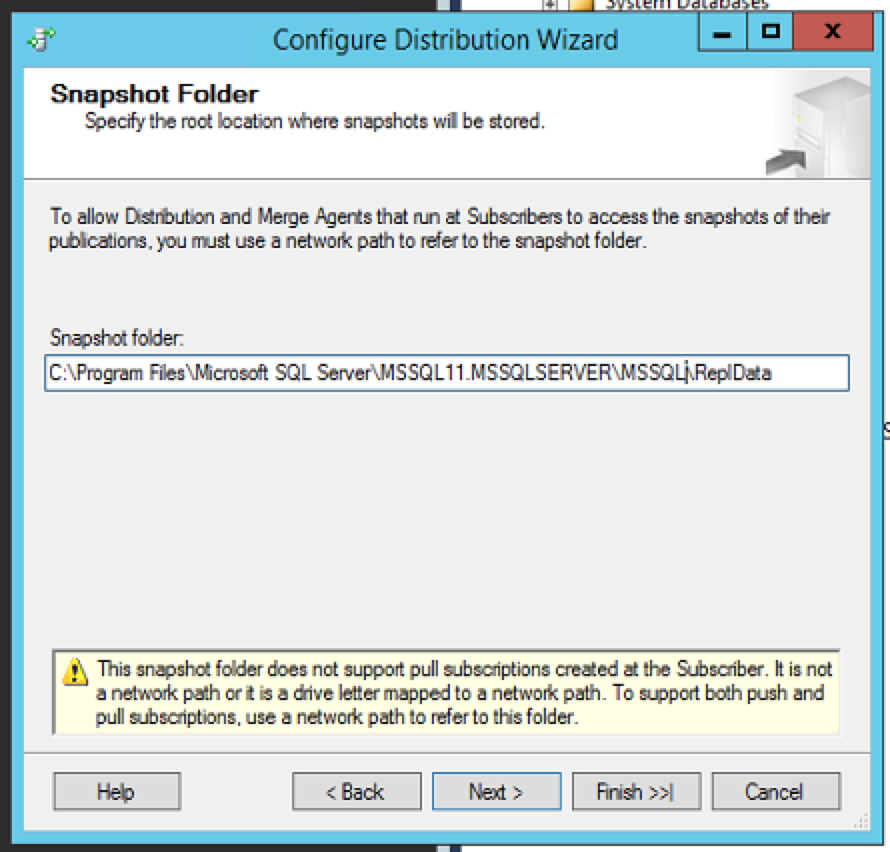
1. Select VEGA-2 as its own Distributor;



1. Create the folder [\\vega-2\ReplData](file:///\\vega-2\ReplData) and refer the Snapshot folder to this share;

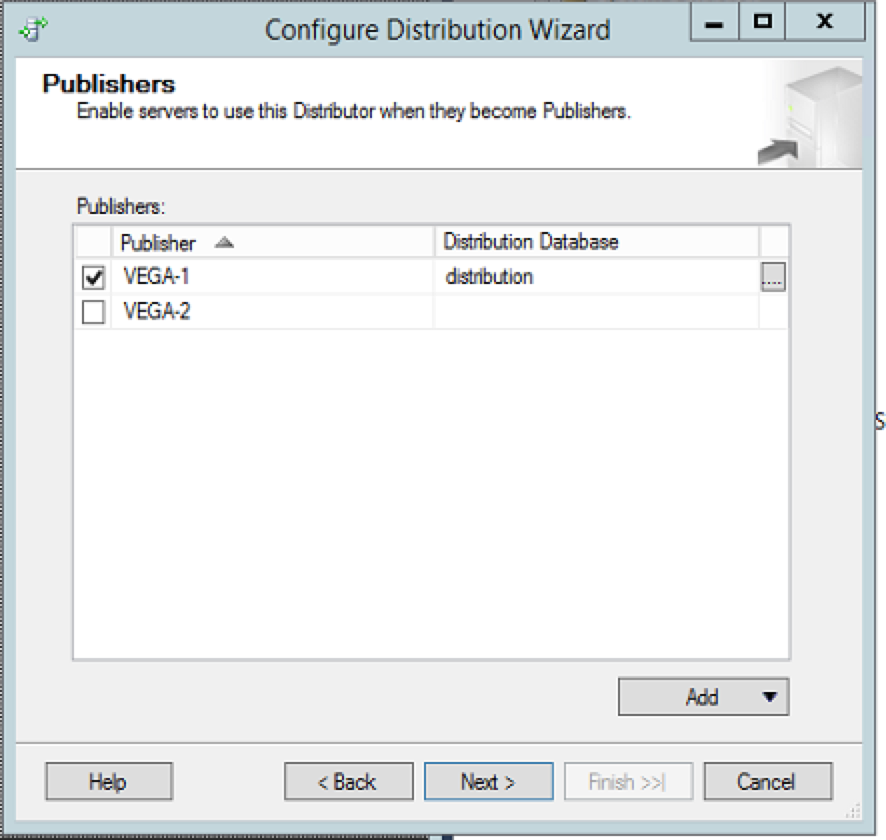


1. Distribution Database settings;

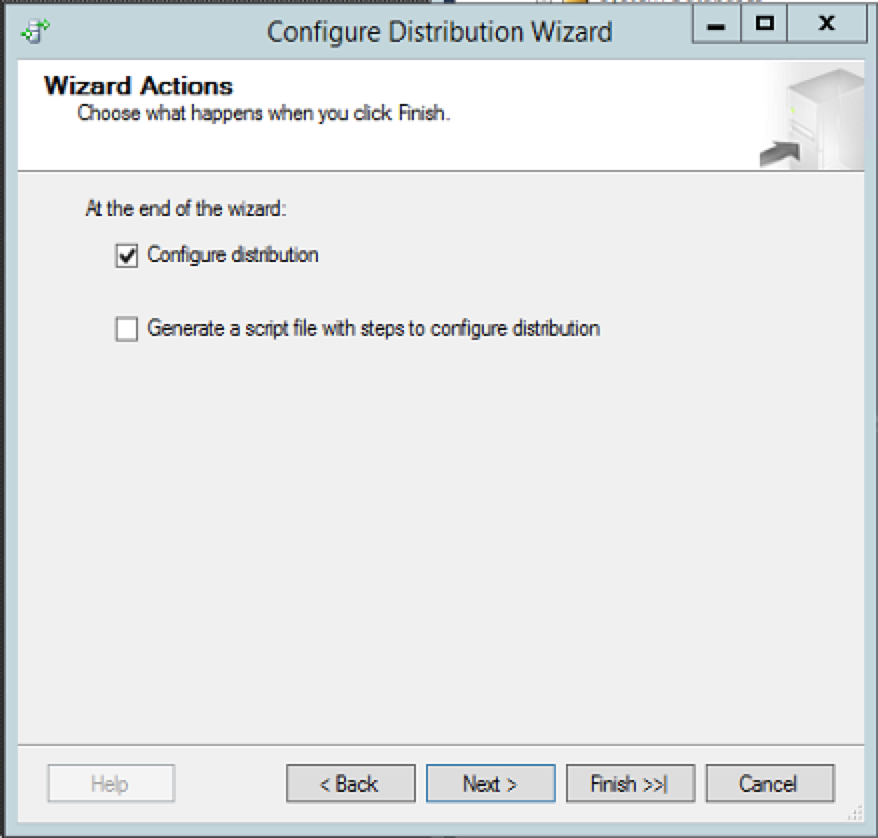


## Obs: Snapshot folder does not support pull subscriptions created at the Subscriber. It is not a network path or it is a drive letter mapped to a network path. To support both push and pull subscriptions, use a network path to refer to the snapshot folder.

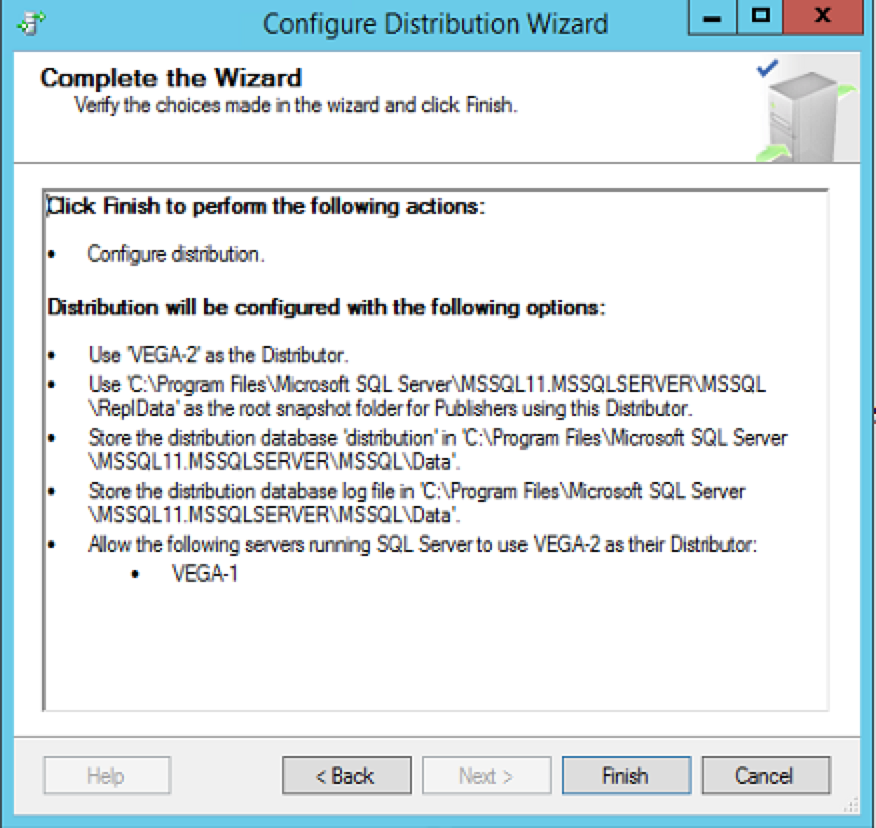
1. Select VEGA-1 as Publisher and uncheck VEGA-2;



1. Create a password (Automatically one login is created in distributor with the name Distributor\_Admin)
2. Check Configure distribution if you want to configure it now and click Next;



1. Finish the wizard to start the Replication tasks;



### Observations

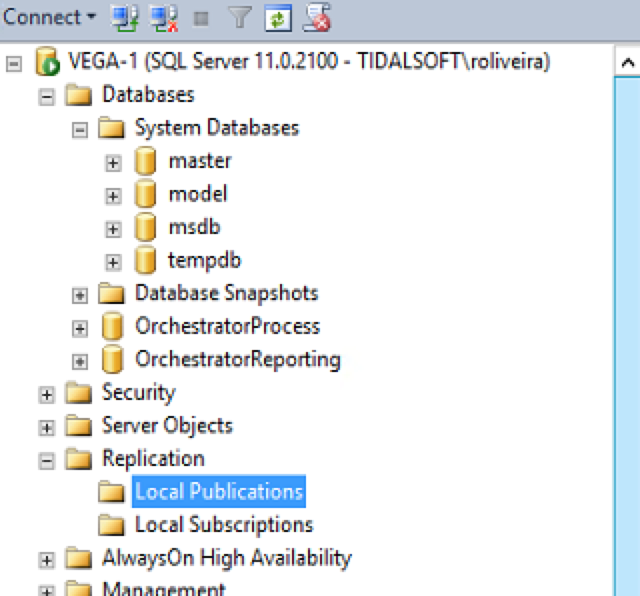
* Go to VEGA-2 -> Databases -> System Databases -> Find the new database “distribution”
* Go to Security -> Logins -> Find a new login “Distributor\_admin”
* Go to Server Objects -> Linked servers -> Find new linked server “repl\_distributor”
* Right Click on Replication -> Select distributor Properties:

## /var/folders/m9/wp8hkpqn5ygcj8n92g11htc80000gp/T/ro.nextwave.Snappy/ro.nextwave.Snappy/Windows 10 x64 VMware Fusion, Today at 1.21.23 AM.png

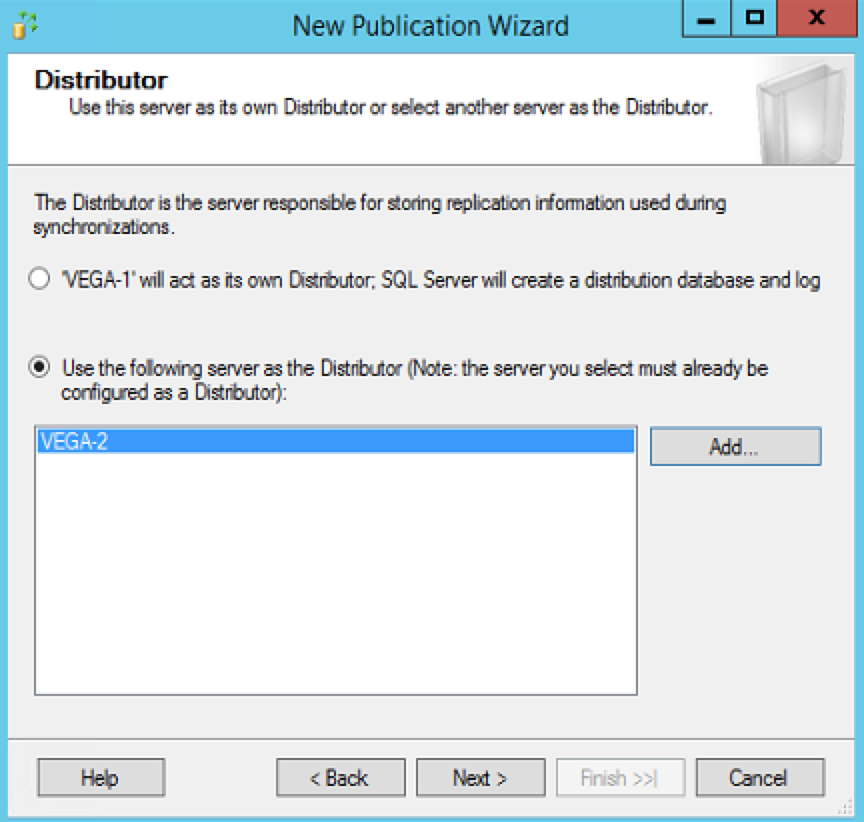
Transactions stored in distribution database are removed after 72 hrs and agents history is removed after 48 hrs.

## Step 2: Creating Snapshot Publication

1. Go to Publisher (VEGA-1) -> Replication -> Right Click on Local Publications -> New Publication.



1. Next;
2. Select ‘Use the following server as the Distributor’ and connect to VEGA-2;



1. Enter the password that you setup in the previous step (for Distributor\_admin) and click Next;
2. Select OrchestratorProcess database;
3. Select the publication type and click Next. Note that we can have 4 types:

* Snapshot publication:

The Publisher sends a snapshot of the published data to Subscribers at scheduled intervals.

* Transactional publication:

The Publisher streams transactions to the Subscribers after they receive an initial snapshot of the published data.

* Peer-to-Peer publication:

Peer-Peer publication enables multi-master replication. The publisher streams transactions to all the peers in the topology. All peer nodes can read and write changes and the changes are propagated to all the nodes in the topology.

* Merge publication:

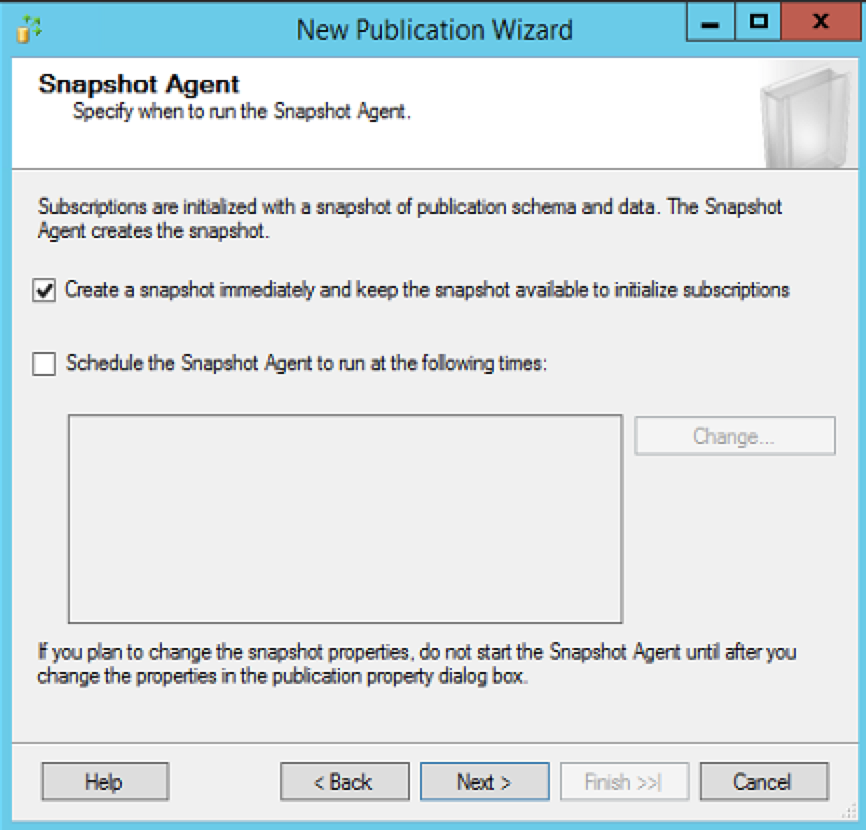
The Publisher and Subscribers can update the published data independently after the Subscribers receive an initial snapshot of the published data. Changes are merged periodically. Microsoft SQL Server Compact Edition can only subscribe to merge publications.

**We are going to proceed using ‘Snapshot publication’ for this example.**

1. Select all articles and click Next;



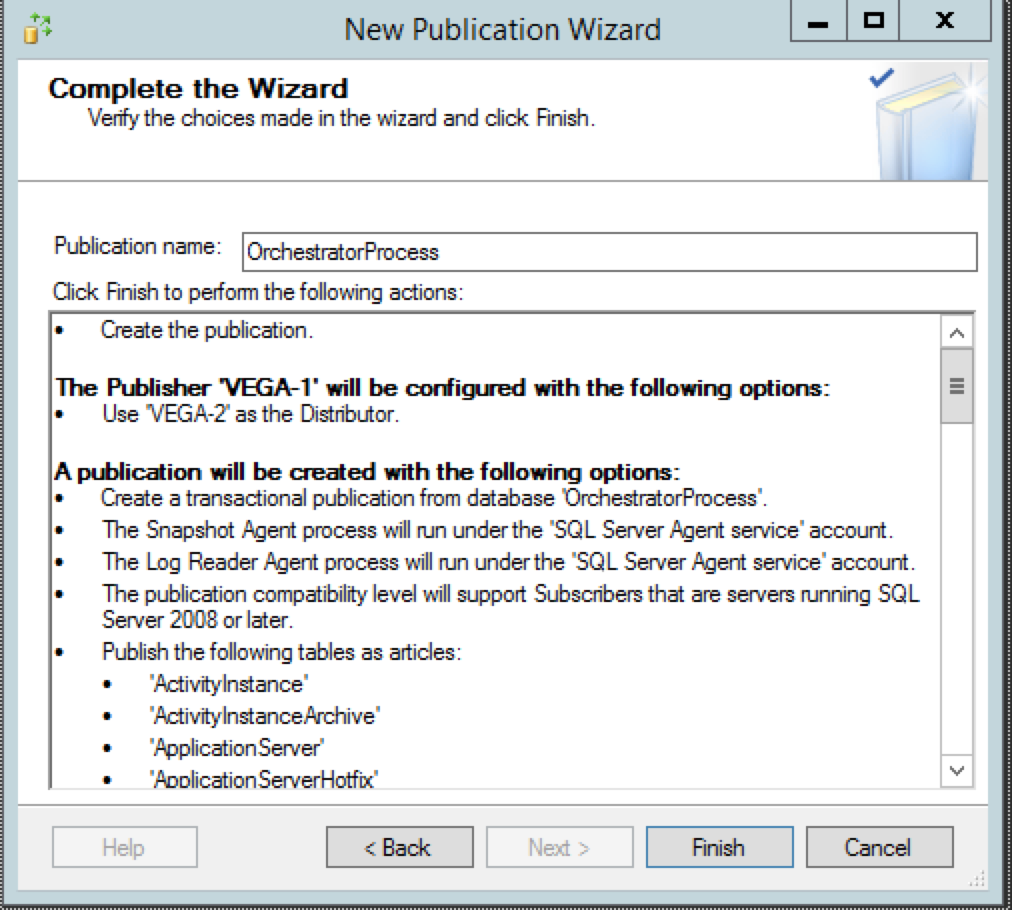
1. In the Article Issues, click Next;
2. On Filters page, click Next;
3. Mark Create a snapshot immediately and keep the snapshot available to initialize subscriptions and click Next;

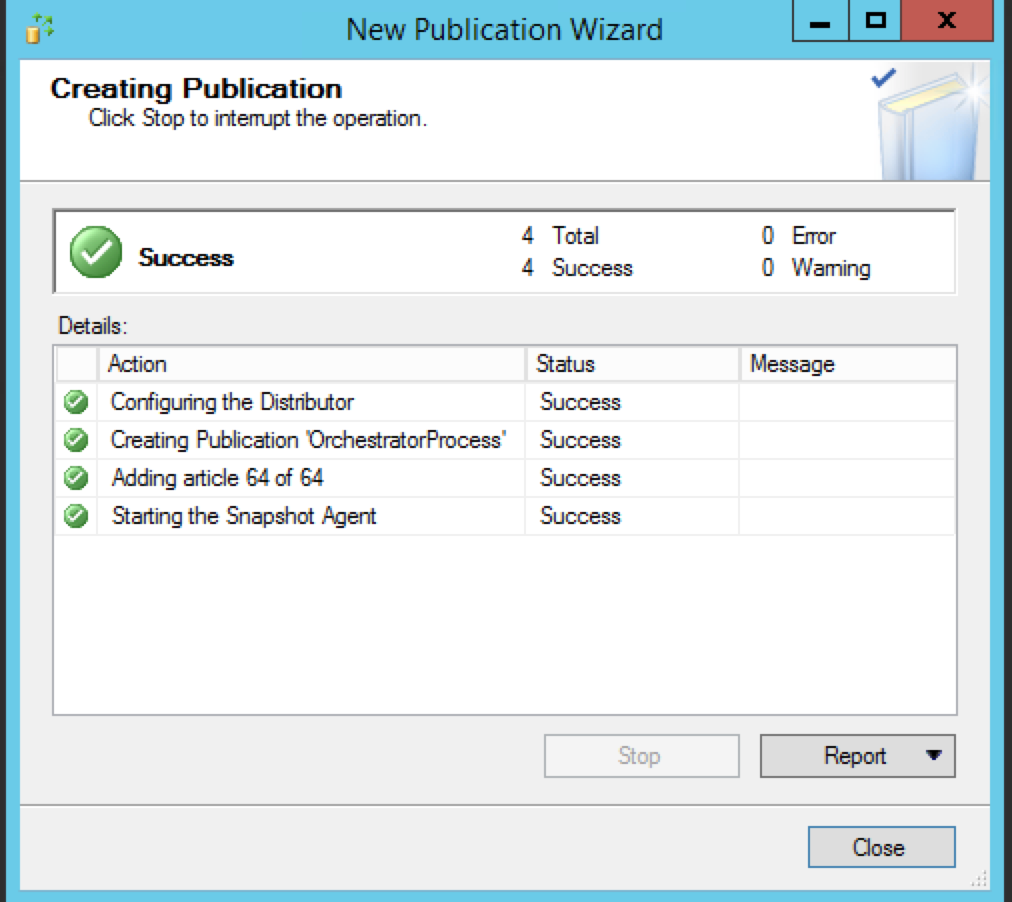


1. Specify the SQL Server Agent service account and impersonate the process account options;

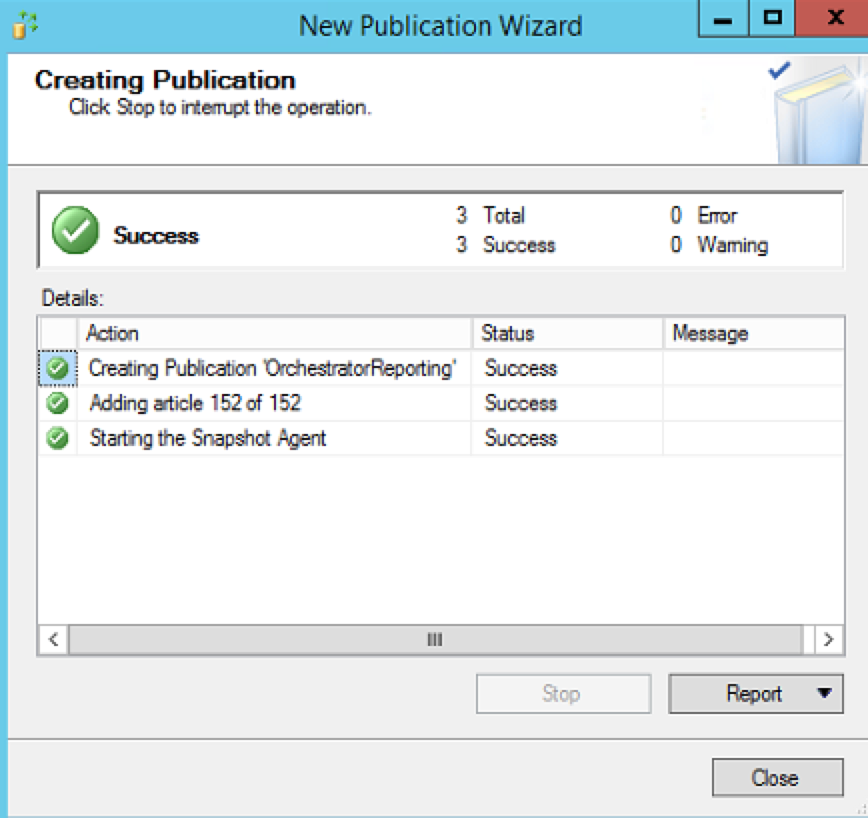


1. Give the publication the name OrchestratorProcess and click ‘Finish’;



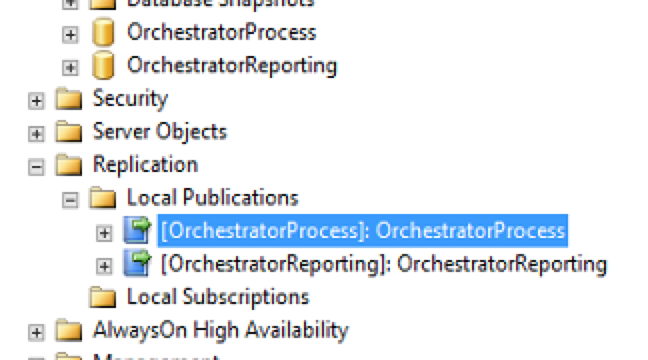


1. Repeat the same steps to publish OrchestratorReporting database.

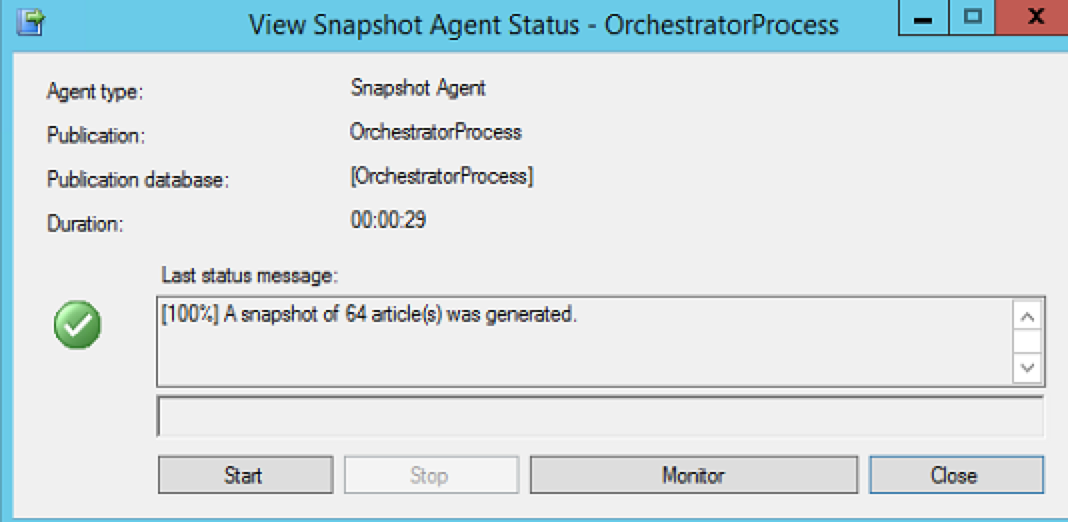


### Observations

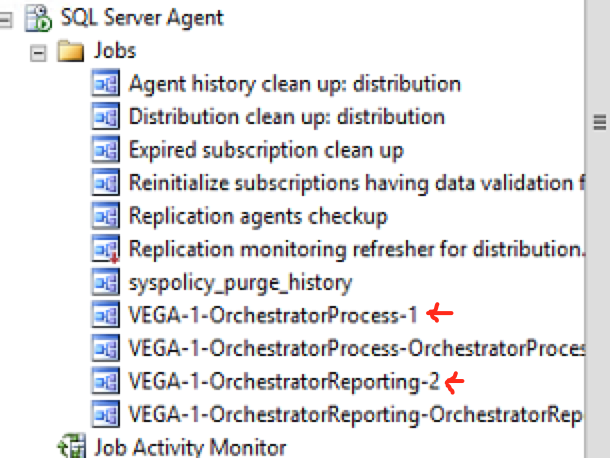
* Go to publisher -> Replication -> Local publications -> See the new created publications;



* Check if snapshot was created properly -> Right click on the publication -> View Snapshot Agent Status;

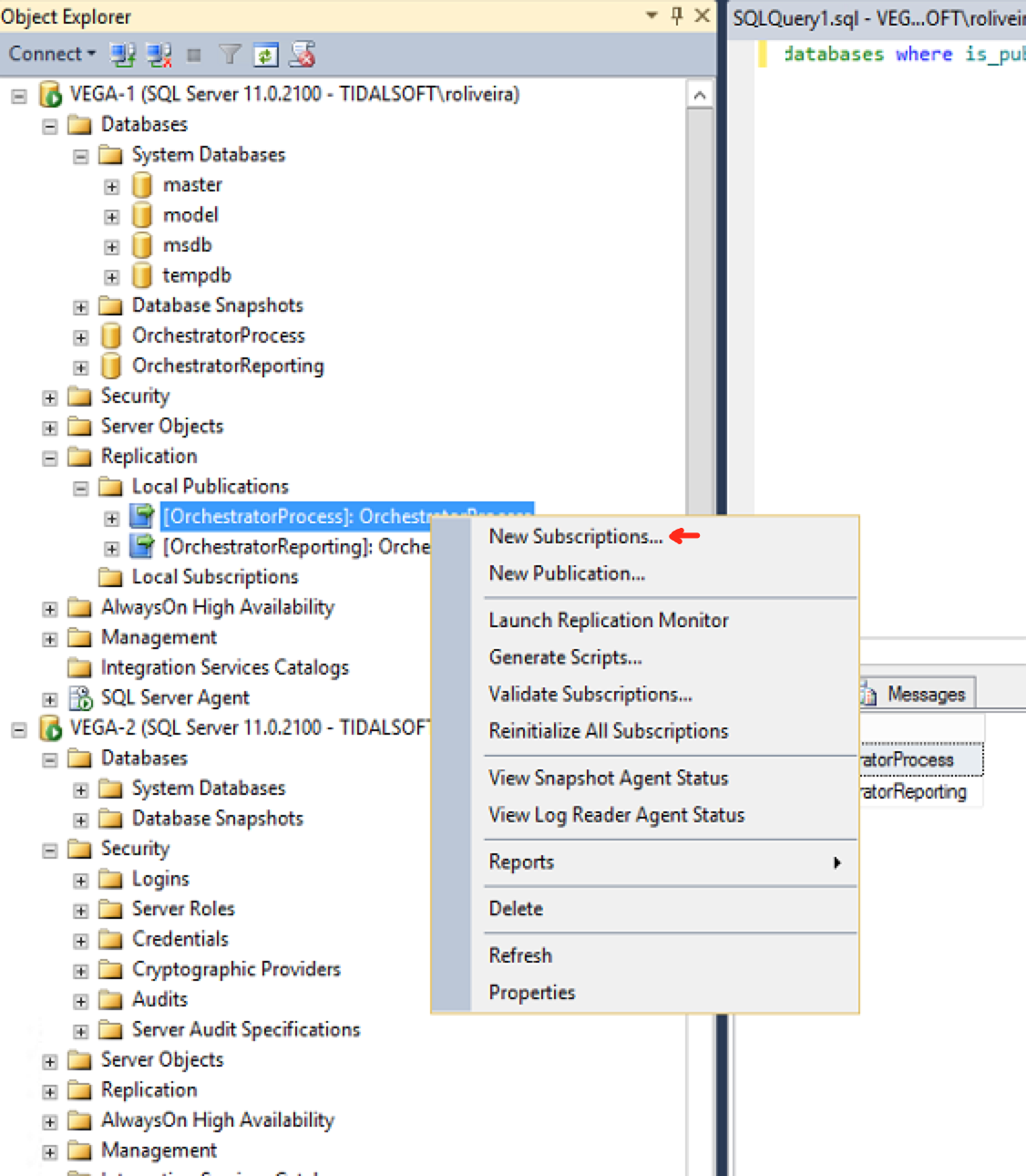


* Go to distributor (VEGA-2) -> SQL Server Agent -> Jobs -> Find snapshot agent job was created;

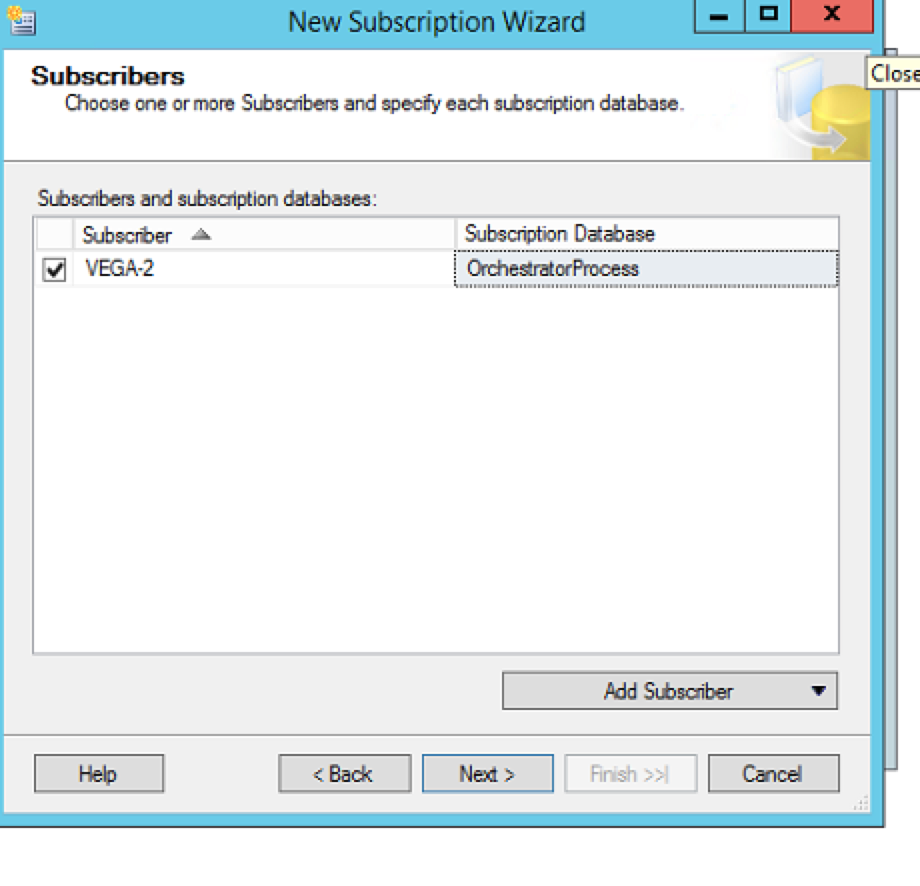


## Step 3: Creating Subscription

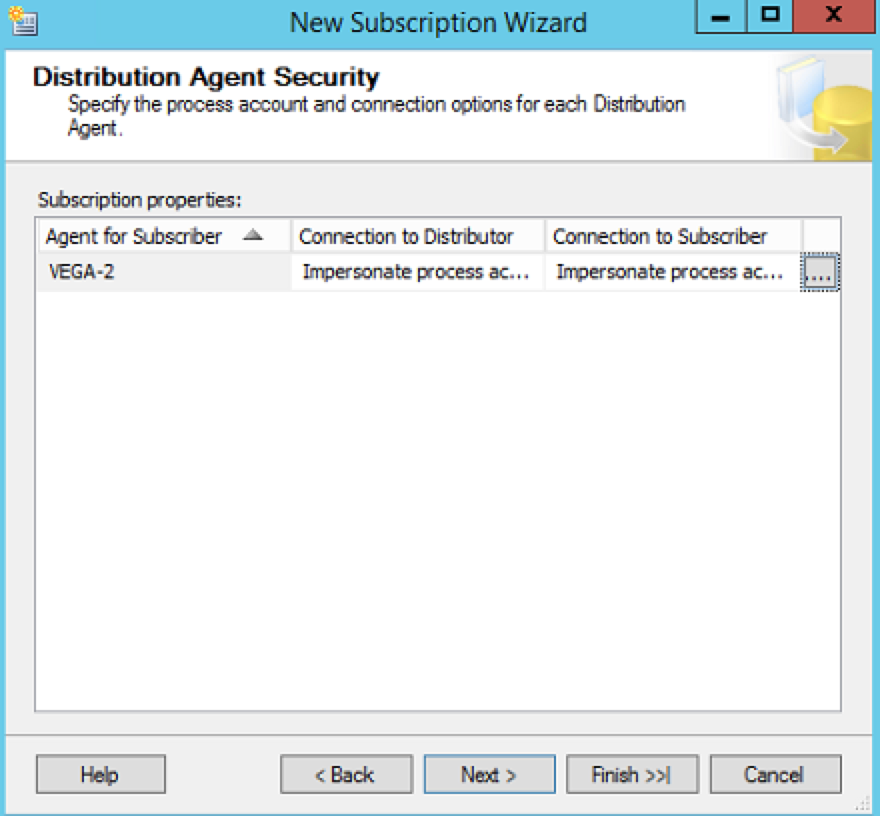
1. Go to publisher (VEGA-1) -> Replication -> Local Publications -> Right Click on OrchestratorProcess -> New Subscription;



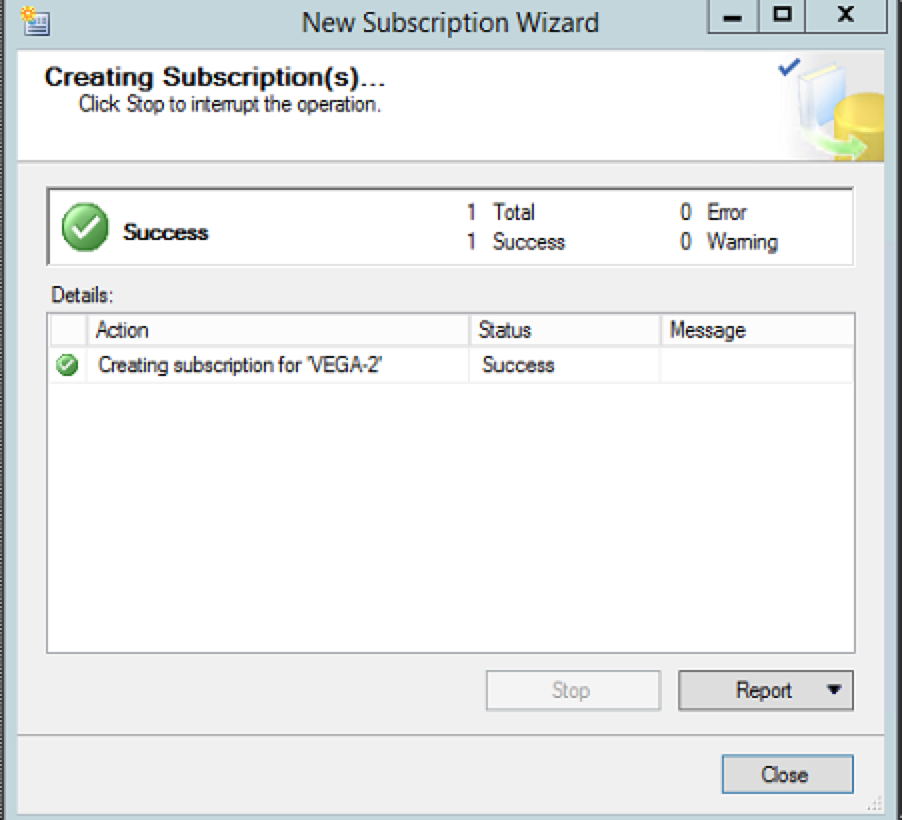
1. Next;
2. Select the publication name OrchestratorProcess and click Next;
3. Select ‘Run all agents at the Distributor (push subscriptions)’;
4. In the Subscribers list, Add VEGA-2 (distributor and subscriber) as Subscriber;
5. Choose the OrchestratorProcess – this should be the Orchestrator Process database from your passive PO install;



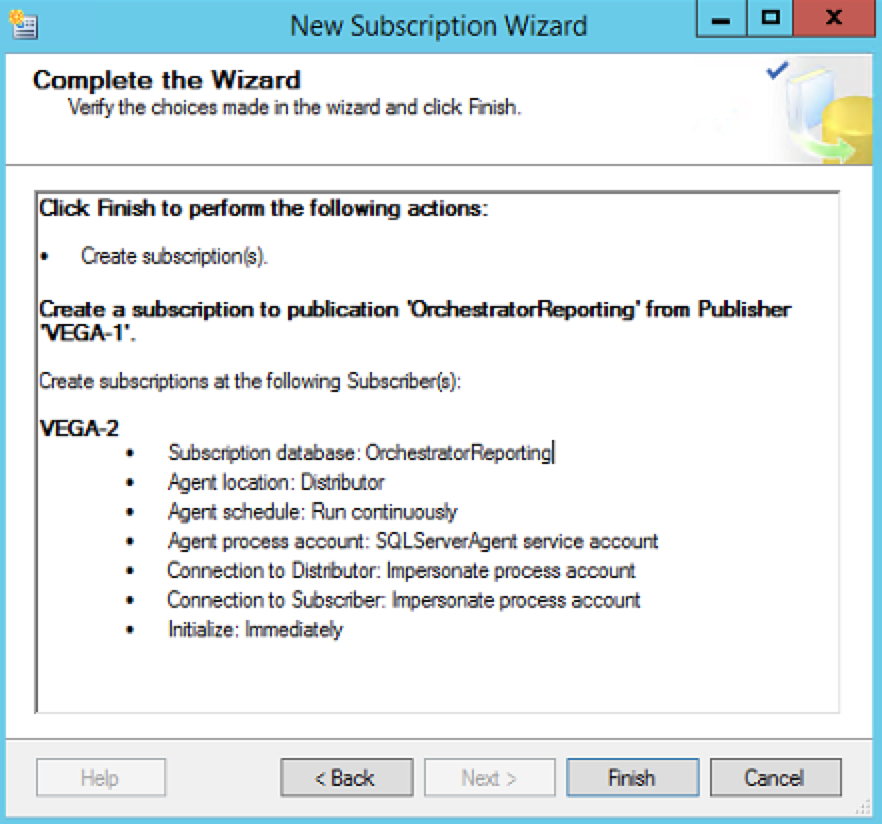
1. Click Next;
2. Click on browse button (…) under distribution agent security page.
3. Select “Run under Agent Service Account” and “By impersonating the process account” options as both distributor and subscriber’s service accounts are the same. If the service account of subscriber is different, create a login in subscriber with sysadmin privileges.



1. Under Agent Schedule -> Select “Run Continuously”
2. Under Initialize when select -> Immediately
3. Next -> Next -> Finish

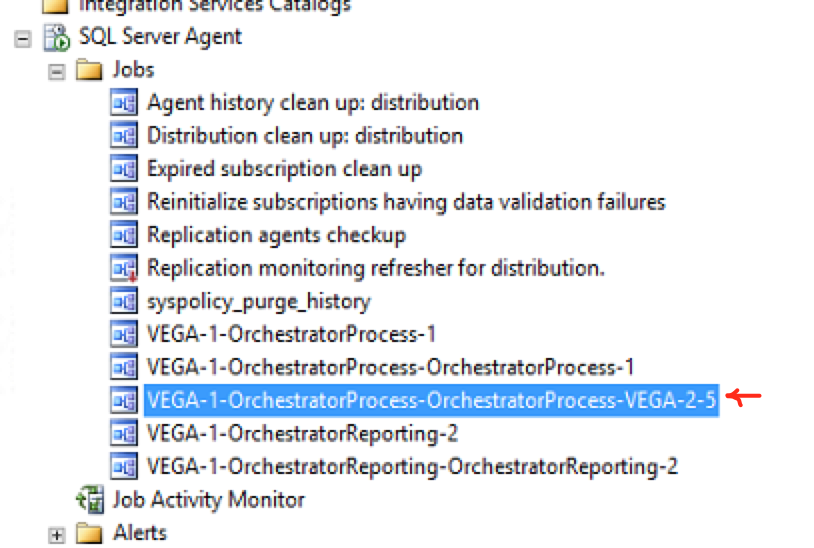


1. Repeat the same steps for OrchestratorReporting database;

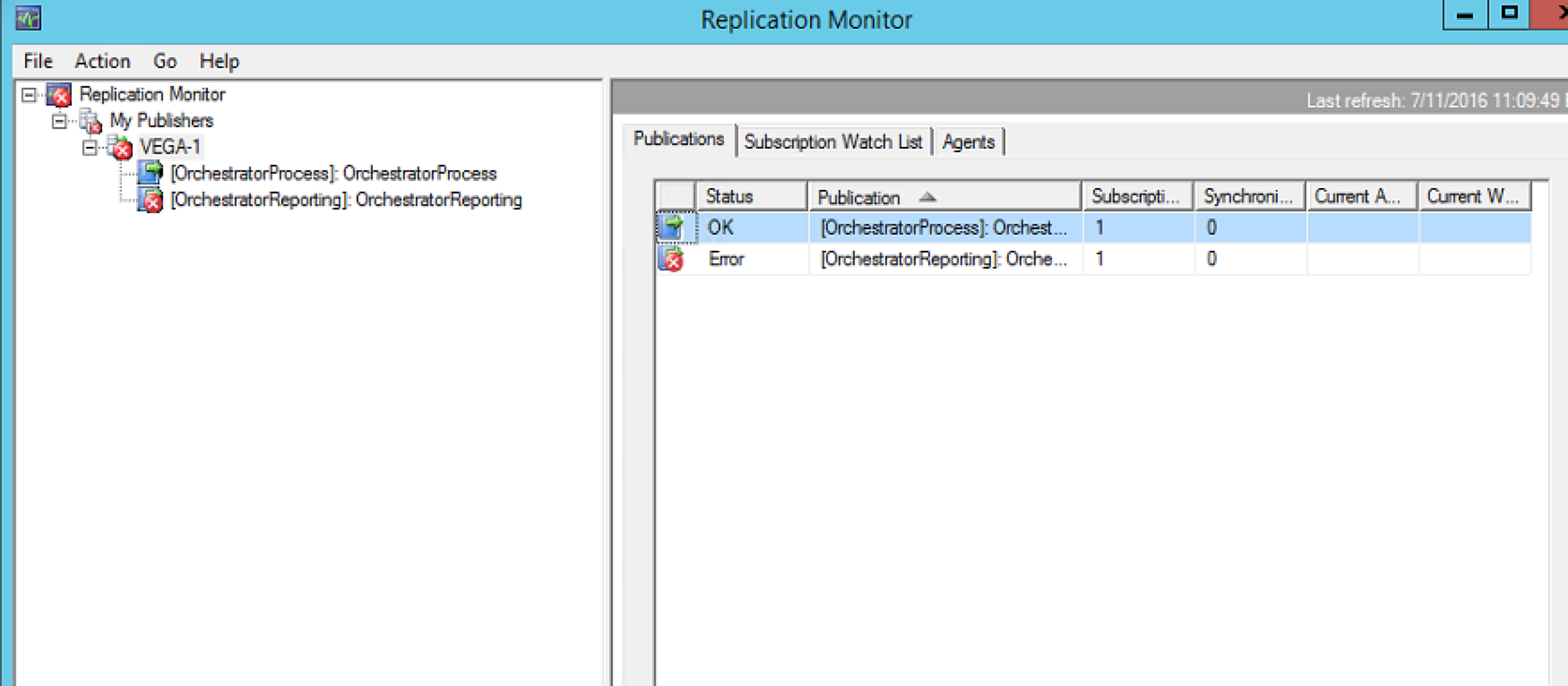


### Observations

* Go to distributor (VEGA-2) -> SQL Server Agent -> Find that a new job was created, related to Distribution Agent;



* From either distributor or publisher server, right click on Replication folder and select Launch Replication Monitor – you can track sync/job issues with this utility;



# Transfer Database Encryption

## Export encryption key to a file using the aspnet\_regiis utility

On VEGA-1, run the following command:

# cd C:\Windows\Microsoft.NET\Framework64\v4.0.30319

# aspnet\_regiis.exe -px "Tidal Intelligent Automation Server" keys.xml -pri

Copy the file

\\Vega-1\c$\Windows\Microsoft.NET\Framework64\v4.0.30319\keys.xml

to

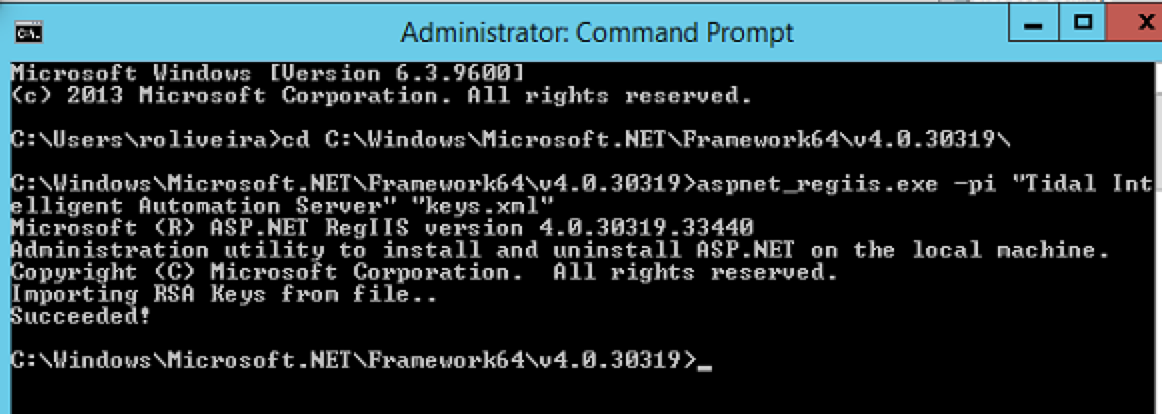
\\Vega-2\c$\Windows\Microsoft.NET\Framework64\v4.0.30319.keys.xml

## Import encryption key from file

On VEGA-2, run the following command:

# cd C:\Windows\Microsoft.NET\Framework64\v4.0.30319

# aspnet\_regiis.exe -pi "Tidal Intelligent Automation Server" "keys.xml"



.Net framework folder location: C:\Windows\Microsoft.NET\Framework64\{version}\aspnet\_regiis.exe

*Might need to specify the full path to this utility if path environment variable is not defined*

# Conclusion

At this point, your Active-Passive PO environment should be configured properly.