Disaster Recovery (formerly High Availability)

Solution Guide

Release: NICE Advanced Process Automation Release 7.1 and 7.2

Document Revision: 7.1 - 7.2 A1
Distribution Status: Published
Publication Date: January 2020

For assistance, contact your local supplier or nearest NICE Customer Service Center:

EMEA Region Tel: +972-9-775-3800 Fax: +972-9-775-3000	**APAC Region** Tel: +852-8338-9818 Fax: +852-2802-1800	**North America** Tel: 1-800-663-5601 Fax: +201-356-2197	**International Headquarters-Israel** Tel: +972-9-775-3100 Fax: +972-9-775-3070
The Americas Region Tel: 1-800-6423-611 Fax: +720-264-4012	**Israel** Tel: 09-775-3333 Fax: 09-775-3000	**France** Tel: +33-(0)1-41-38-5000 Fax: +33-(0)1-41-38-5001	**Hong-Kong** Tel: +852-2598-3838 Fax: +852-2802-1800
United Kingdom Tel: +44-8707-22-4000 Fax: +44-8707-22-4500	**Germany** Tel: +49-(0)-69-97177-0 Fax: +49-(0)-69-97177-200		

For more information about NICE, visit www.nice.com

CONTENTS

1

Introduction

Advanced Process Automation supports Disaster Recovery (formerly known as High Availability) using an active and a passive site to allow business continuity.

The transition from one site to the other is by manual failover.

This guide describes the tasks that are required to set up Advanced Process Automation for Disaster Recovery, the maintenance and failover procedures, and the expected impact on the solution.

The Disaster Recovery solution is built as follows:

1. DNS Alias or Public Load Balancer – Ensures that RTI clients and users work with the active site and are redirected to the passive site upon failover by the DNS Alias or Public Load Balancer that they are connected to.
   - For attended solutions (RTPO, RTAM and Engage), the client is not usually installed in the data center, and would use a DNS Alias or Public Load Balancer.
   - For Robotic Automation the client is usually installed in the data center and therefore additional Robotic Automation clients are needed on the passive site.

2. Data is replicated between the active and passive sites using Microsoft DFS to make sure up-to-date data is available upon failover:
   - SVN
   - ActiveMQ
   - Elasticsearch

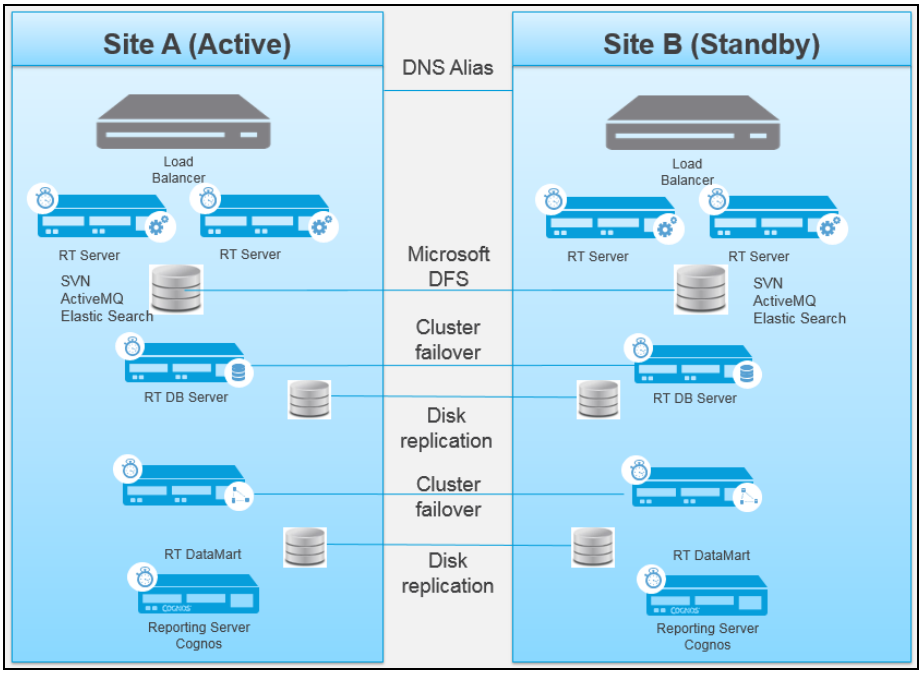
3. For database resiliency, two options are available:
   - MS SQL failover cluster database and disk replication
     - RTS Databases (Operational, DataMart, and Cognos repository) files are replicated using Microsoft DFS
   - SQL AlwaysOn

4. RTS Databases are implemented on top of an MS SQL failover cluster database.

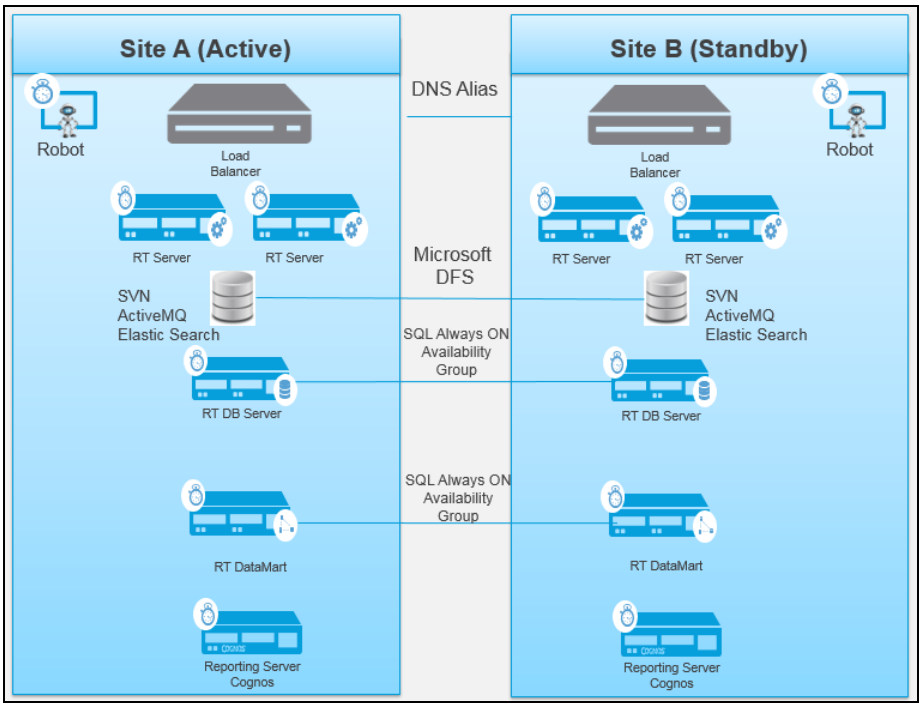
5. The Cognos server works in an active/standby mode.

This guide refers to the following architecture:

Cluster Failover - Disk Replication



SQL AlwaysOn



Document Revision History

Revision	Description
7.2-A1 Advanced Process Automation 7.2 January 2020	■ Added link for RT Server Installation in Cluster environment to Workflow: Creating a Disaster Recovery Environment on page 10. ■ Moved the topic **Performing a Disaster Recovery Failover Procedure** after topic **Setting Up an APA Disaster Recovery Environment**. ■ Added scenarios to test the Disaster Recovery environment. See Testing the Disaster Recovery Flow on page 93. ■ Updated Configuring Cognos Disaster Recovery on page 82 and Limitations and Prerequisites on page 18.
7.2-A0 Advanced Process Automation 7.2 December 2019	■ Added Setting Up a Secured Clustered Database TLS 1.2 Environment on page 84. ■ Added manual steps to Performing a Disaster Recovery Failover Procedure on page 89. ■ One shared, cross-release version for 7.1 and 7.2

[This page intentionally left blank]

2

Workflows

Select and follow the correct workflow.

Contents

Workflow: Creating a Disaster Recovery Environment

Follow this workflow to create an APA Disaster Recovery environment. For details, see Setting Up an APA Disaster Recovery Environment on page 23.

Task	Description	Reference
1.	Review the limitations and prerequisites.	Limitations and Prerequisites on page 18
2.	Configure SQL Server disaster recovery. Create the Windows cluster, database servers, and the database environment to support SQL AlwaysOn.	Configuring SQL Server Disaster Recovery (AlwaysOn APA Deployment) on page 24
3.	Create the DNS Alias\Public Load Balancer to support the forwarding of the inbound load balancers of the sites.	Configuring DNS Alias Disaster Recovery on page 47
4.	Install the replication system, for example Microsoft DFS Replication, to support file replications for the following: ▪ SVN ▪ Elasticsearch ▪ ActiveMQ	Setting Up DFS Replication Groups on page 51
5.	Configure multiple Cognos Content Manager components to support disaster recovery.	Configuring Cognos Disaster Recovery on page 82
6.	Install the RTServer at the active site (Site A). Point the server to the one of the physical databases while installing it.	*Installing a Real-Time Server* in the *Real-Time Server Installation and Upgrade Guide*. When installing the server as a cluster node, you must verify that the Load Balancer machine is configured or the installation will fail.
7.	Change the RTServer to point to the listener of the Availability Group (AG).	

Task	Description	Reference
8.	Stop the services on Site A.	
9.	Install the RTServer at the passive site (Site B), pointing it to the SQL Server AlwaysOn (AO) listener.	*Installing a Real-Time Server* in the *Real-Time Server Installation and Upgrade Guide.* When installing the server as a cluster node, you must verify that the Load Balancer machine is configured or the installation will fail.
11.	At site A, configure all replicas for SVN, Elasticsearch, and ActiveMQ.	▪ Configuring SVN Disaster Recovery on page 78 ▪ Configuring Elasticsearch Disaster Recovery on page 80 ▪ Configuring ActiveMQ Disaster Recovery on page 81
10.	After the site is up and tested, set up a secured connection for a clustered database.	▪ Setting Up a Secured Clustered Database TLS 1.2 Environment on page 84 a. Securing the Database Servers in a Clustered Environment on page 84 b. Securing the Cognos Server on page 87 c. Securing the Real-Time Server on page 87

Workflow: Performing a Disaster Recovery Failover

Follow this workflow to perform a Disaster Recovery failover. For details, see Performing a Disaster Recovery Failover Procedure on page 89.

Task	Description	Reference
1.	Review the limitations and prerequisites.	Limitations and Prerequisites on page 18
2.	Validate the environment: ▪ Databases are working ▪ AlwaysOn is working ▪ Storage servers are available and running ▪ All services are running.	Performing a Disaster Recovery Failover Procedure on page 89
3.	Move to use the passive site: ▪ Stop the services at the active site ▪ Change the DNS Alias to point to the passive site ▪ Start the services at the passive site	Performing a Disaster Recovery Failover Procedure on page 89

3

Preparing for Disaster Recovery

Review this section before creating and configuring the APA Disaster Recovery environment.

Contents

Advanced Process Automation Resiliency

This section describes the components in an Advanced Process Automation system that are involved in Disaster Recovery.

Real-Time Server

The Real-Time server hosts the following main components:

Components	Description
Apache Web server	Web server used for communication with the clients.
Policy agent	Enforces the access policy to the server.
Application server Apache Tomcat	Hosts the application: Gets client status. Collects data from the clients and stores it in the database. Imports users into the server. Manages the solution publishing and assignment to clients.
OpenAM access management	Manages the server authentication policy configuration and decisions.
Elasticsearch	Used for the Automation Portal - Robotic Client Control Room and Task Control Room.
ActiveMQ	Message broker used by the Application server to manage incoming messages queue from the Real-Time clients.
SVN	Software versioning and revision control that is used for APA solution version control. Solution files are retrieved from the Real-Time server and loaded by Real-Time clients.
Vault	The Vault manages the encryption and decryption of the passwords used in the Real-Time server.

Components	Description
Cognos server	Used for reporting (Cognos may be installed on a dedicated server according to the Design guidelines).
Microservices	Microservices are independent modular services where each service runs a unique process and communicates through a lightweight mechanism to serve a business goal. Examples in APA include the Centralized Credential Management, Application Monitoring, and Password Store microservices.

Real-Time Server Clustering

The Real-Time server cluster works in an active-active mode. The Real-Time server clustering solution aims to provide high availability. It uses application clustering, meaning that the cluster members are independent Windows machines (and not a Windows cluster), each with its own application installed and configured so they are mutually aware of one another.

All Real-Time server resiliency solutions are based on load balancing, either hardware based or software based. The load balancer distributes the client-server SOAP communication load between application server nodes/cluster members. The Real-Time Client saves the request locally and does not delete it until it gets a successful response.

Real-Time Server nodes synchronize between themselves. The Real-Time Server keeps a local cache of which solutions exist, solution assignments to teams and the configuration from the configuration table, and so on. When one of these is changed on one server, it must inform the others of the change to verify that they are all aware of the new state.

The load balancer is hosted on a dedicated server or appliance.

SVN Clustering

The Apache SVN server provides versioning control for the product solutions and projects. These solutions and projects are saved to the SVN server (hosted on the Real-Time Server) using the SDS Microservice , where they can be versioned when necessary. The Real-Time Clients access the RTServer which allows using the SDS Microservice the download of the user corresponding solution, from the SVN. When several Real-Time Servers are deployed (for scale or resiliency), each Real-Time Server has a local SVN on top of it. One SVN is defined as the master SVN and other SVNs are defined as replicated. Publish and download can be done from any node as long as the master is available.

The SVN replication is performed when a new solution is published to the repository on any change in the repository.

Queue Management and Distributed Cache (for Robotic Automation)

ActiveMQ supports the queue that stores the robot requests and controls the robots. It includes the following queues:

- Inbound queue: Stores new automation requests to be pulled by the robots.

- Outbound queue: Stores processed automation results.
- Control In queue: Incoming messages from a robot to the server. For example, when a robot pulls a new automation from the In-Queue, it sends a message to the server using the Control-In-Queue.
- Control out queue: Outgoing messages from the server to a robot. For example, when the server instructs the robot to restart.

In addition, the product includes In-progress cache, which is a fast-access persistent cache to manage requests on the server used for reducing the load on the database.

In a cluster, ActiveMQ is deployed for load sharing to allow execution of tasks by robots to other nodes.

Elasticsearch

Elasticsearch is used as a search engine for information that needs to be presented in real time. Elasticsearch is used to store

- RTI client information such as files version and assigned solutions,
- Robotic tasks and exceptions
- Desktop Automation tasks

Elasticsearch in a cluster is deployed in high availability mode to enable redundancy when one node fails.

APA Databases

Several databases are used by an APA system:

- **Real-Time Server database:** Stores the system and user information. Data from the clients is stored in the database (data collection and activity monitoring).
- **Data Mart database:** Extracts data from the operational database for reporting and long term retention.
- **Automation Finder database:** Stores the Automation Finder data.
- **CCM database:** Stores data used by Centralized Credential Management (CCM).
- **Troubleshooting database:** Data gathering database used for troubleshooting.

The Disaster Recovery solution supports Microsoft SQL databases. For Microsoft SQL database resiliency, the solution supports the local Microsoft clustering solution.

Real-Time Client

The Real-Time client holds all the logic and intelligence for the solution. After the business logic is defined using the Real-Time Designer or Automation Studio, it is published to the SVN in the Real-Time server and is then downloaded by the Real-Time client. The Real-Time client uses the business logic to collect the defined data, application monitoring data, and process monitoring data, and to perform automation, etc.

To monitor the Real-Time client process, the RTI Launcher service is installed on the Agent desktop/Citrix server. This service makes sure the Real-Time client is up and running. If the client crashes or goes down, RTI Launcher makes sure the Real-Time client is restarted on the agent desktop or in the same Citrix session.

If the Real-Time client loses connectivity to the Real-Time Server, it uses store and forward mechanism. It buffers all the data in the local cache and then pushes the data to the Real-Time server when it is available. The size of the data that can be buffered by the Real-Time client is only limited by the hard disk space available on the agent desktop.

During the Real-Time server failure, the Real-Time client used the solution files already downloaded, only if the user had already connected to the server in the past and the solution file exists in the **AppData** folder. During this time the Real-Time client is not able to download a new version of the project file, if the new version exists.

In some scenarios the Real-Time server and database have to be available in order to ensure the solution works as expected.

The scenarios include the following: Using Real-Time server as proxy for web services or database query.

Robotic Automation Client

The Robotic Automation client consumes a solution file that is created by a power user using the Real-Time Designer.

Solution files are retrieved from the Real-Time server and loaded by the Robotic Automation client.

Once loaded, the Robotic Automation client is then configured and ready to execute the Robotic Automation solution business rules.

The Robotic Automation client pulls automation requests from the queue and reports back to the server.

Limitations and Prerequisites

Before setting up and using Disaster Recovery, you must verify that the APA site environment meets all requirements and take into account the caveats and limitations shown below.

Caveats and Limitations

1. SVN – The slave node is a read only repository, which means that a new solution is not published until the master node is restored. SVN must be updated at all nodes in parallel. This is done by replicating the SVN repository, for example using the Microsoft DFS Replication system.

2. ActiveMQ – ActiveMQ queues are replicated in order to keep current messages in the queues.

3. Database replication - For best results, the replication group should be updated online so that the group always contains the updated sources.

Caveats and Limitations for Robotic Automation

Event	Function	Impact on Local Cluster	Impact on Failover between Sites
Server goes down	Automation Request on the server	Requests are available in the database. ▪ Requests are restored to queue when queue restarts from disk. ▪ If server is lost, they are marked as backlogged and can be re-invoked by API	Requests are available in the database. ▪ They are marked as backlogged and can be re-invoked by API
Server goes down	Robot that processes request	Yes, continues seamlessly and reports to the other server	Request continues because all queues are replicated (ElasticSearch, ActiveMQ)
Server goes down	Elasticsearch – task view	Automation Requests available on the second node and continues seamlessly	Automation requests are available in the passive site due to the replication

Prerequisites

> **NOTE:** The SQL AlwaysOn prerequisites are covered in Configuring SQL Server Disaster Recovery (AlwaysOn APA Deployment) on page 24. DNS Alias for database and Disk Replication for database are not required for the SQL AlwaysOn database.

Domain Prerequisite

Domain environment is required for Disaster Recovery. Both Data Centers and all NICE components must be in the same domain.

The replication group should be updated online so that the group always contains the updated sources.

Disk Replication Requirements

Disaster Recovery supports storage array-based replication. After configuring disk replication; verify that it is configured properly.

These are the general requirements for disk replication for storage-array based replication:

- The disk replication is crash consistent.
- The disk replication solution is near real-time.
- For each data center, there must be a dedicated RAID group configured on the SAN for each of the following:
  - SQL data – sizing should be determined by NICE design documentation
  - SQL logs - sizing should be determined by NICE design documentation
  - SVN repository – the repository should be at least 10GB
- The SVN repository should be mounted locally as drive s: (on both sites).

For each RAID group, there is a dedicated Logical Unit Number (LUN). The LUNs are replicated during the disk replication procedure.

To verify disk replication:

There is no hard set of rules for verifying disk replication, and the exact procedure varies across vendors.

However, the verification should include the following main steps:

1. On the source disk on Machine 1, create a temporary directory and files.
2. Replicate the disk from Machine 1 to Machine 2.
3. Verify that the data created in Step 1 exists on the destination disk on Machine 2.
4. Change the direction of the disk replication.

5. On Machine 2, edit data and replicate the data from Machine 2 to Machine 1.

6. Verify that the data changes appear on Machine 1.

DNS Alias Prerequisites

Disaster Recovery uses DNS alias names for the following Real-Time Server components:

- Application Server - In a Disaster Recovery environment, only one instance of the Applications Server runs at a time. When configuring the DNS alias, you create one DNS alias name (CNAME) in the DNS server. The DNS alias name initially points to the network name of the Load Balancer Server on Data Center 1.
- SVN repository – The DNS alias name points to the network name of the Load Balancer (in order to unify the resource URL described in the DB).
- Database server - The DNS alias name points to the virtual hostname of the SQL cluster group of this cluster on Data Center 1.
- Cognos server – The DNS alias name points to the network name of the Cognos server.

NICE recommends that you modify the Time to Live (TTL) for the DNS alias name to 5 minutes.

Modifying the TTL for DNS records

This section describes the steps to modify the Time to Live (TTL) on DNS records. TTL is used by name servers and some DNS clients to determine the length of time that a name must be cached. The default TTL for DNS CNAME records is 60 minutes, and Disaster Recover configuration requires the TTL to be 5 minutes. Use the DNS Management console to modify the TTL of the record.

By default, the TTL field is not displayed on records. To display the TTL field on records, enable the **Advanced View** feature.

For more information on modifying the TTL, see http://support.microsoft.com/kb/297510.

Verify that the required DNS aliases are defined before completing this procedure.

To modify the TTL for DNS records:

1. In Administrative Tools, open the DNS Management console. Click **View**.

2. Select the **Advanced** check box.

3. Right-click the record that you want to modify. The record displays a TTL field that you can modify.

4. In the **Time to live (TTL)** field, clear existing values and enter 5 in the MM (minutes) part of the field.

5. Repeat steps 1 to step 5 for each DNS record that is required by Disaster Recovery.

6. After creating the DNS aliases, it is required to test that the same IP address appears for the alias name and the host name of the server for which the DNS alias was created. It is recommended to check both host names for each alias.

[This page intentionally left blank]

4

Setting Up an APA Disaster Recovery Environment

This chapter describes how to set up an APA Disaster Recovery environment.

Contents

Configuring SQL Server Disaster Recovery (AlwaysOn APA Deployment)

APA with AlwaysOn provides a high-availability and disaster-recovery solution for SQL Server. It makes use of existing SQL Server features, particularly Failover Clustering, and provides new capabilities such as availability groups.

This section explains how to deploy APA with AlwaysOn Availability Groups.

Prerequisites

Make sure the following prerequisites are met:

- If the datamart database and operational databases are installed on the same server, they must be on the same instance.
- The paths of the database data files, the database log files and the backups must be identical in the Primary node and the secondary nodes. Also, if the product contains the datamart database , an identical path for the SSIS package which populates the datamart must be created on each node of the availability group of the DataMart.
- If RTI products are installed on the same SQL Server instance as NIM/Engage, all SQL logins and Windows logins, which have been manually created or altered on the primary replica, must be manually created or altered on the secondary replicas as well. Their security settings must also be copied to all secondary replicas. The logins are not synchronized

> **NOTE:** For a full list of Microsoft prerequisites for the SQL AlwaysOn availability groups (SQL Server) solution, see the relevant Microsoft software documentation.

Limitations

- The secondary replica must be defined as readable replica, otherwise the synchronization job (synchro job) is not able to run on the secondary replica's node, and the environment is not synchronized.
- When AlwaysOn is installed on a node that hosts Engage/NIM as well as RTI, the synchronization is executed by the Engage/NIM synchro job. In this case, logins are not synchronized. Note that Engage/NIM does not support automatic failover.
- Deletion of a login is not synchronized to the secondary replica's node, because RTI has no way to know whether the login is mapped to a user in a database not connected to RTI product. Deletion of logins from the current secondary's' nodes must be done manually.
- Not all jobs are replicated to the secondary replicas' nodes, only jobs that follow naming conventions: If the synchronization is handled by a Nice RTI Syncro Job, only jobs with names beginning "Nice RTI ..." are synchronized to the secondary nodes. If the synchronization is done by Engage's/NIM's job only jobs with names beginning "Nice..." are synchronized.
- Not all backup devices are synchronized. If the synchronization is done by the RTI job, only jobs with a name that begin with "NICE RTI..." are synchronized. If the synchronization is done by Engage's or NIM's job, only jobs with a name that begins with "NICE.." are synchronized.

- If the synchronization on the secondary replica is done by an RTI job, all jobs with names beginning "Nice RTI" are disabled. If the synchonization on the secondary replica is handled by an Engage/NIM job, all jobs with names beginning "Nice" are disabled. As a result, if more than one replica resides on a SQL instance, all the replicas must have the same role, either all primary or all secondary. You cannot have some replicas as primary replicas and others as secondary replicas. This limitation is true only for replicas that contain databases from Nice. It is recommended to have all databases from Nice on the same replica.

Creating Data, Log and Packages Partition

On every SQL machine that is part of the availability group, create drives for data, logs, backups and SSIS package (If datamart is installed). The drives must be identical in each availability group.

Installing APA Products

Install the APA product, specifying the SQL Server IP address for the SQL Server.

Add NICE APA Databases to Availability Group

It is the customer's responsibility to install, configure, backup, and maintain the SQL databases used by APA. The database are:

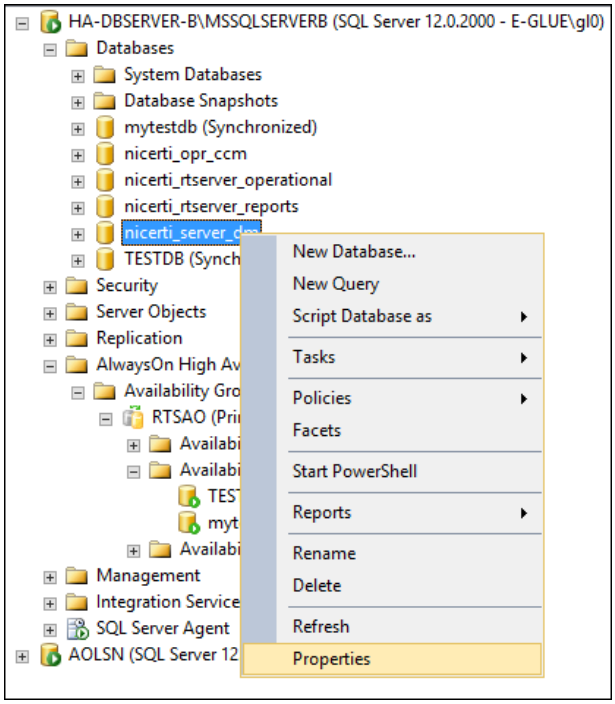
- **Real-Time Server database:** Stores the system and user information. Data from the clients is stored in the database (data collection and activity monitoring).
- **Data Mart database:** Extracts data from the operational database for reporting and long term retention.
- **Automation Finder database:** Stores the Automation Finder data.
- **CCM database:** Stores data used by Centralized Credential Management (CCM).
- **Troubleshooting database:** Data gathering database used for troubleshooting.

This procedure shows how to change NICE databases to a full backup model, and then back them up. The NICE databases are then added to the Availability Group.

Important! Only one High Availability group should exists for nicerti databases on the instance. This means that nicerti databases should always be added to the existing High Availability group. There should not be separate groups for each database.

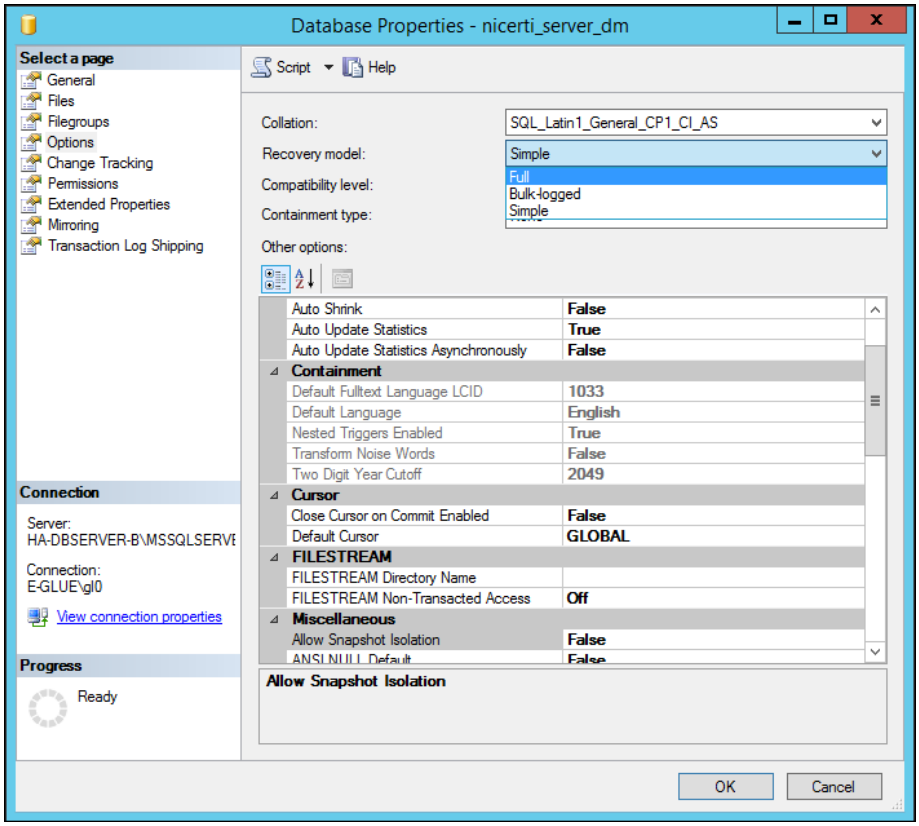
To change NICE APA databases to a full backup model:

1. Open SSMS and connect to the Primary replica.
2. Go to **Databases**, and click the NICE database. Each NICE APA database starts with **nicerti_***. In the example, it is **nicerti_server_dm**.
3. Right-click the database, and click **Properties**.

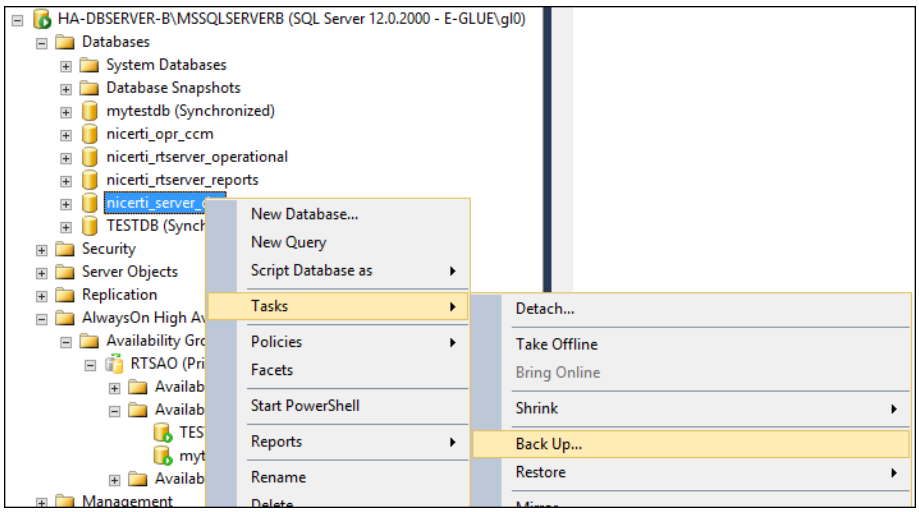


4. In the left panel, click **Options**.

5. Change the Recovery model from **Simple** to **Full**. Click **OK**.

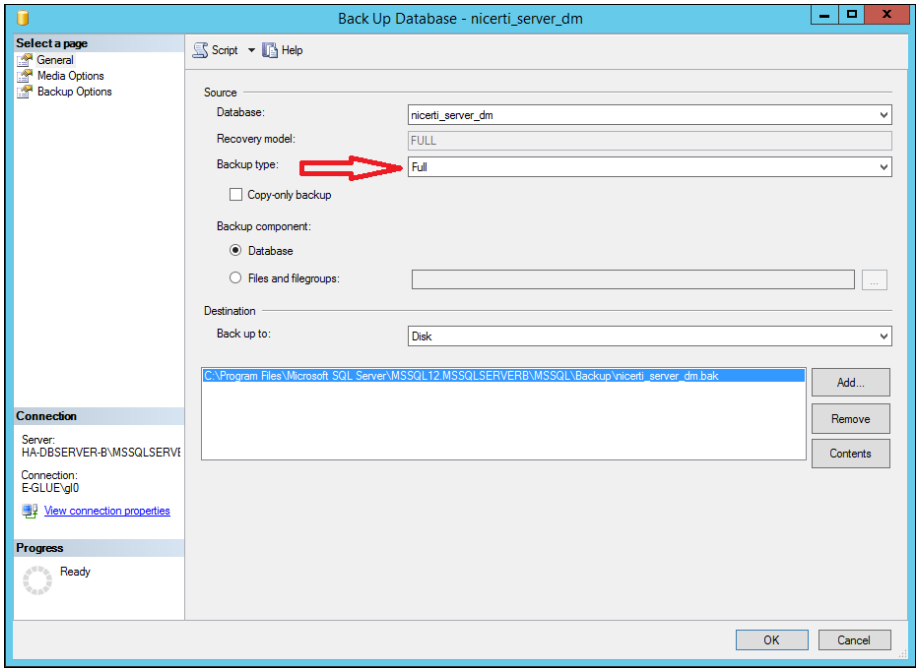


6. In the Object Explorer, expand the **Databases** folder. Right-click a nicerti_* Database folder and select **Tasks →Back Up**.

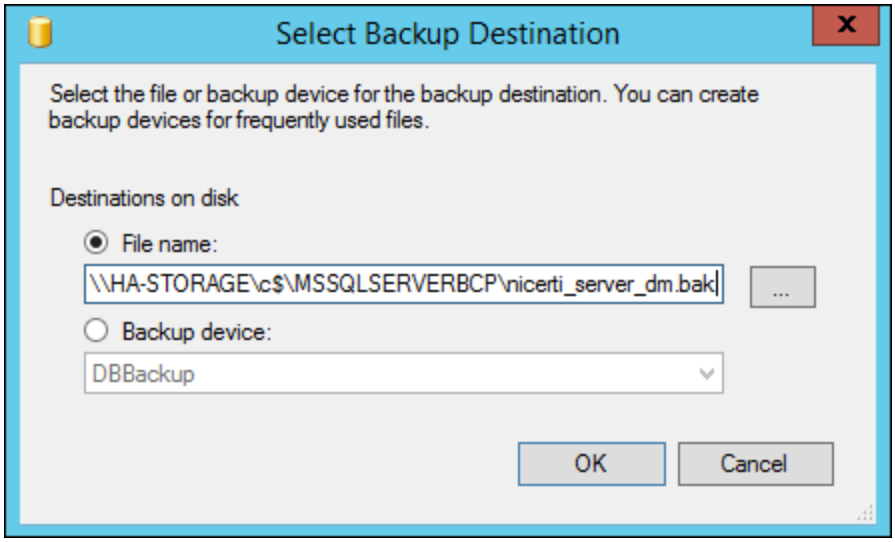


The Back Up Database Wizard window appears.

7. Verify that **Backup Type** is **Full**.



8. In **Destination**, select **Disk** and click **Add**.



The Select Backup Destination wizard appears.

9. In **Destinations on Disk**, enter a file name for the backup the database located in a shared folder that is accessible from both Data Centers. Click **OK**.

10. Repeat steps 1 to 9 for all **nicerti_*** databases.

11. Click **OK** again to start the Database Backup procedure.

The nicerti_databases can now be added to the Availability Group. Continue below.

To add a nicerti database to an existing Availability Group:

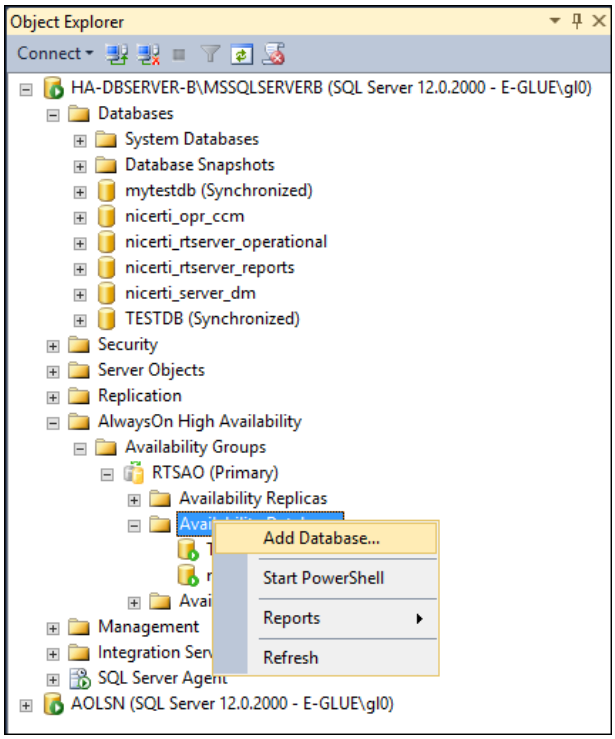
Important! Only one High Availability group should exists for nicerti databases on the instance. This means that nicerti databases should always be added to the existing High Availability group. There should not be separate groups for each database.

1. Open SSMS and connect to the Primary replica.

2. In the Object Explorer, go to **AlwaysOn High Availability** and click **Availability Groups**.

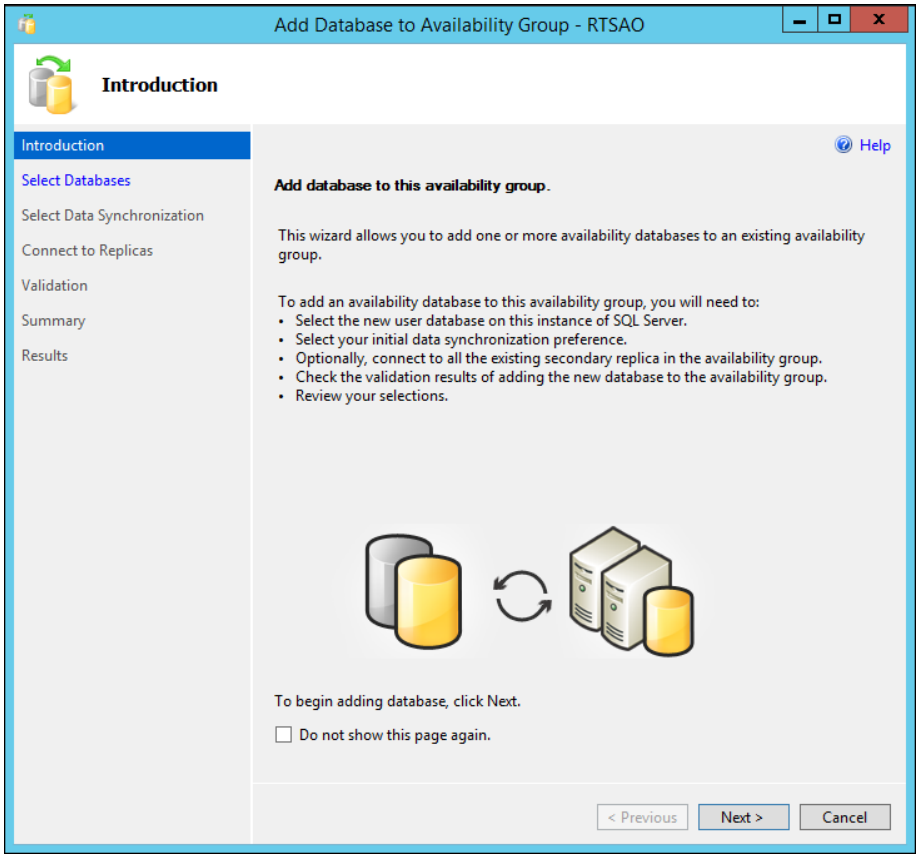
3. Select the name of the Availability Group and expand it.

4. Right-click **Availability Databases**, and click **Add Database...**

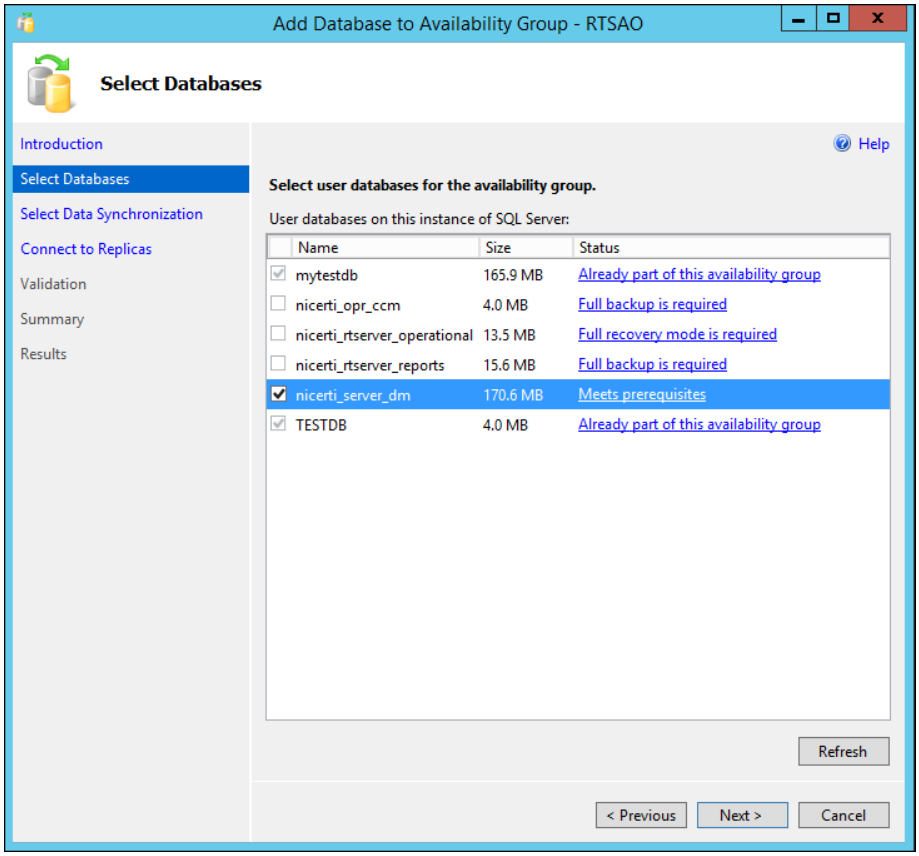


The Add Database to Availability Group wizard appears.

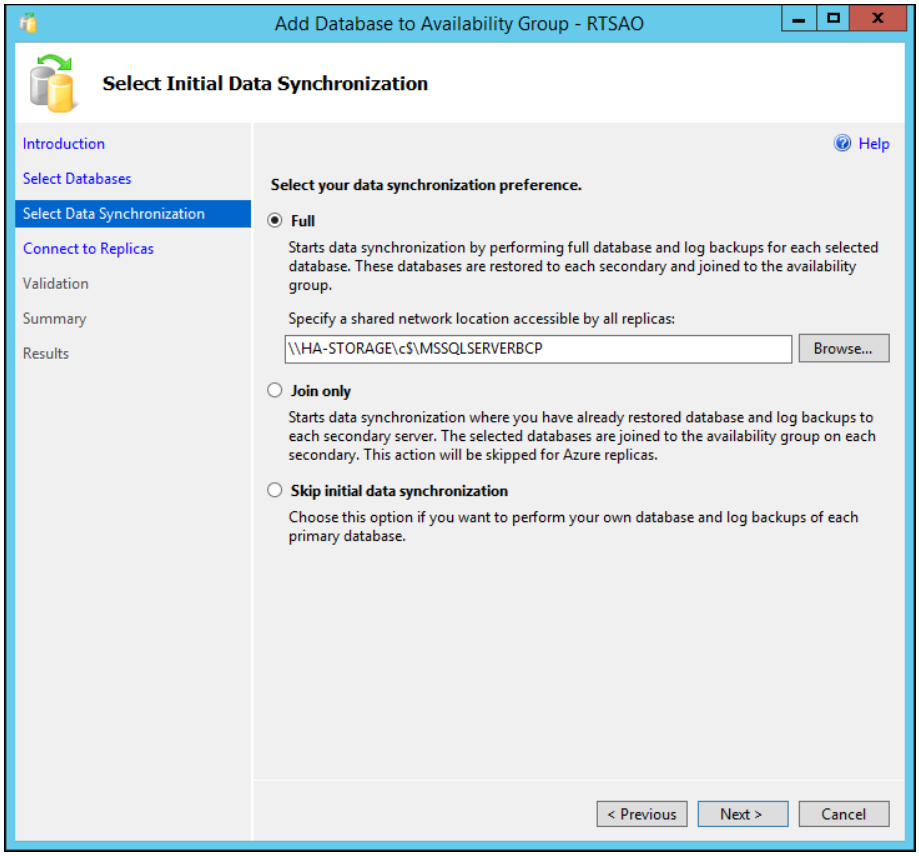
5. Click **Next**.



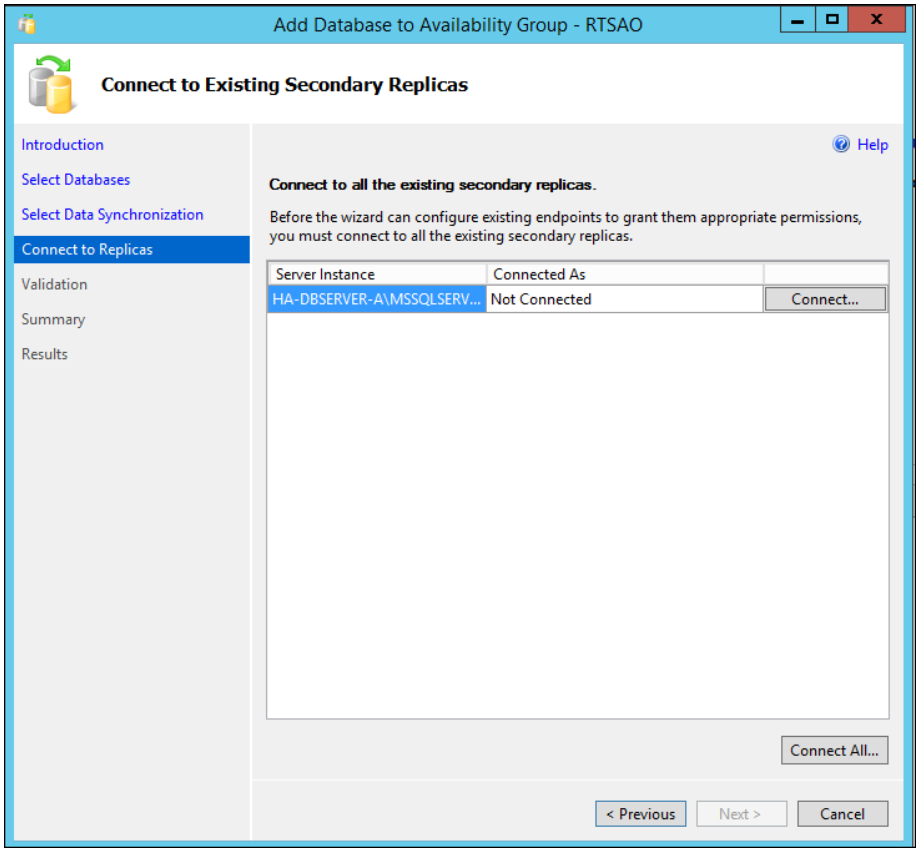
6. Select the NICE databases and click **Next**.



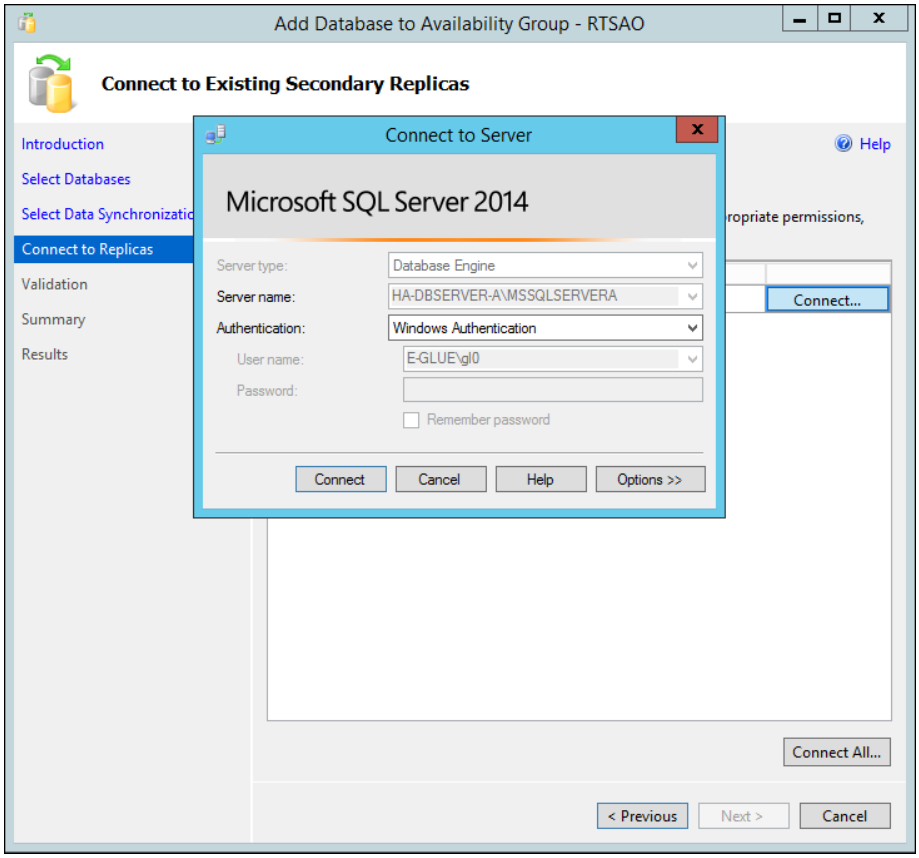
7. In Select Data Synchronization, select **Full**, enter the shared location folder and click **Next**.

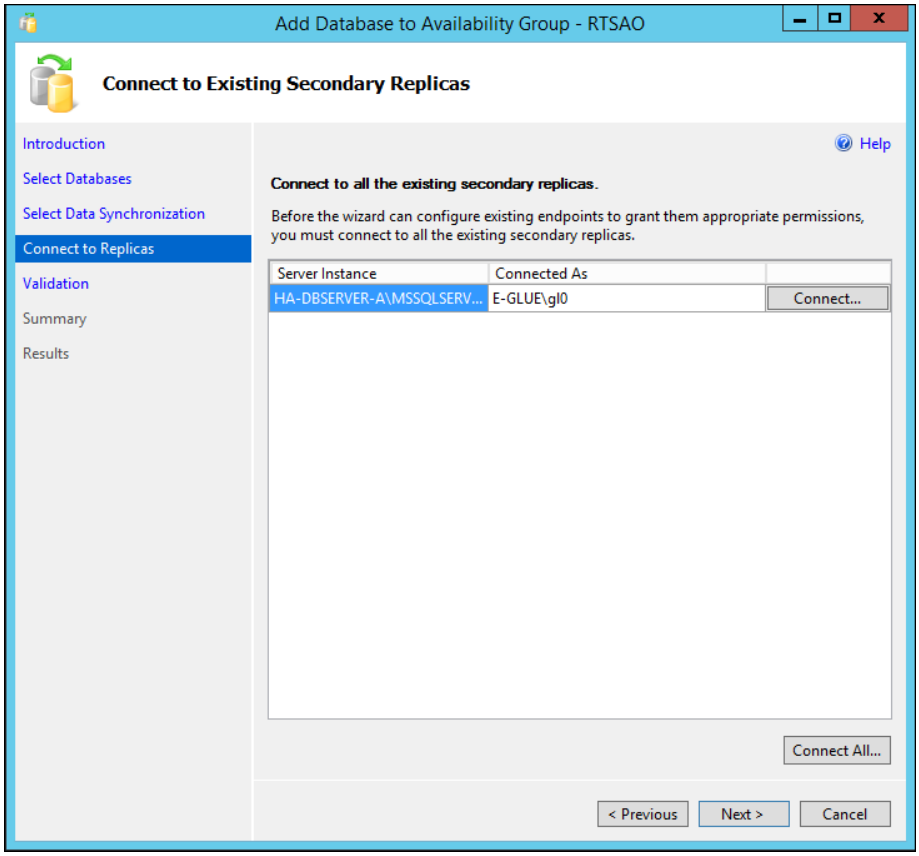


8. In Connect to Replicas, click **Connect**.



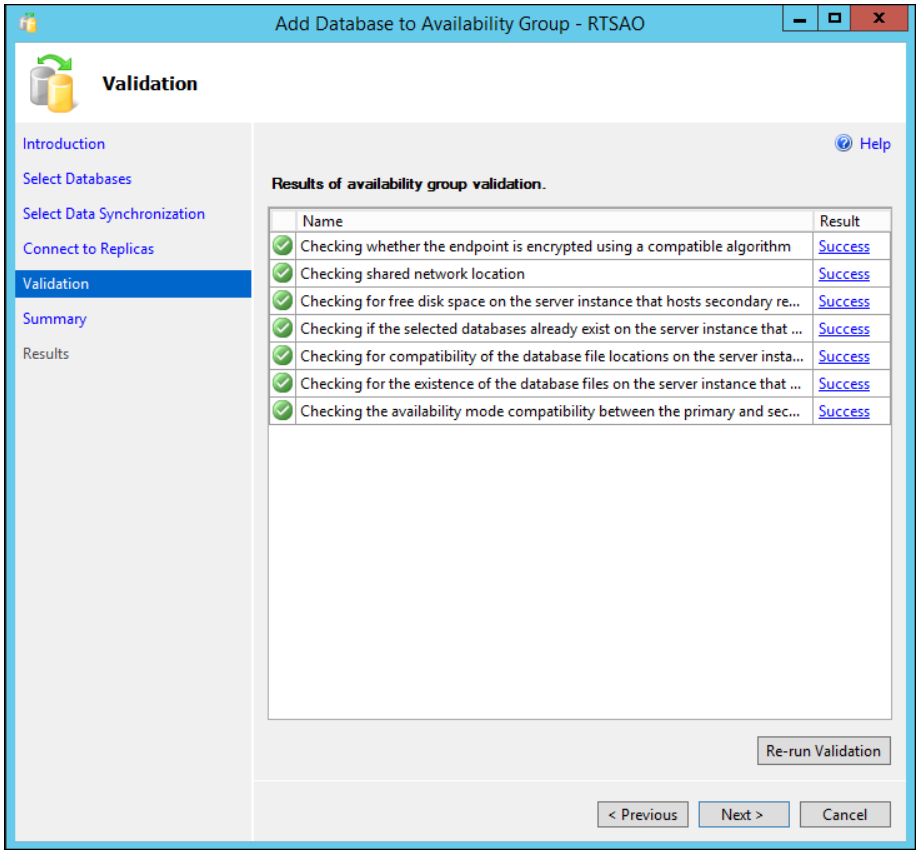
9. In the **Connect to Server** window that appears, enter your server details.


Add Database to Availability Group - RTSAO
Connect to Existing Secondary Replicas
Introduction
Select Databases
Select Data Synchronizatic
Connect to Replicas
Validation
Summary
Results
Help
ropriate permissions,
Connect...
Connect All...
< Previous
Next >
Cancel
Connect to Server
Microsoft SQL Server 2014
Server type:
Database Engine
Server name:
HA-DBSERVER-A\MSSQLSERVERA
Authentication:
Windows Authentication
User name:
E-GLUE\gl0
Password:
Remember password
Connect
Cancel
Help
Options >>



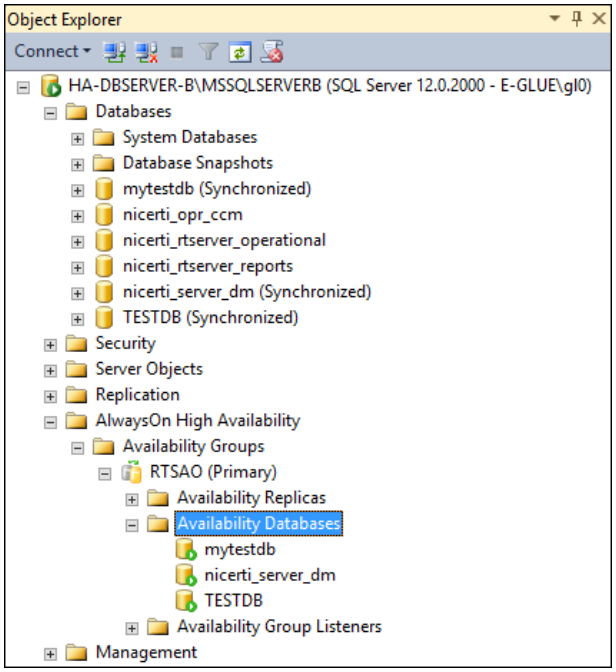
10. When connected, click **Next**.

11. Follow the instructions in the Validation window to complete availability group validation.



12. Repeat the above steps of each one of nicerti database.

13. In SSMS verify the following:
    - All nicerti databases appears under **Availability Databases**
    - Under **Databases**, all nice databases appears as **Synchronized**



14. In a clean installation, backup the service master key from the primary database (master) and then restore it on the secondary database (slave):

    a. On the primary database backup the key by running this command:

    ```
    BACKUP SERVICE MASTER KEY TO FILE = '\\<Path to Master Key\filename>' ENCRYPTION BY PASSWORD = '<password>';
    ```

    <Path to Master Key\filename> is the complete path including file name of the Service Master Key to be backed up. This must be a network location accessible from both the primary and secondary sites.

    <password> is the password that will be used for restoring the Service Master Key.

    Example:

    BACKUP SERVICE MASTER KEY

    TO FILE = 'h:\temp_backups\keys\service_master_key'

    ENCRYPTION BY PASSWORD = '3dH85Hhk003GHk2597gheij4';

    b. On the secondary database restore the key from backup by running this command:

    ```
    RESTORE SERVICE MASTER KEY
    FROM FILE = '\\<Path to Master Key\filename>'
    DECRYPTION BY PASSWORD = '<password>';
    ```

    <Path to Master Key\filename> is the location and filename of the Service Master Key backup in Step a.

<password> is the password that was used when backing up the Service Master Key.

Example:

RESTORE SERVICE MASTER KEY

FROM FILE = 'h:\temp_backups\keys\service_master_key'

DECRYPTION BY PASSWORD = '3dH85Hhk003GHk2597gheij4';

15. During an upgrade, when adding an encrypted database, the Master Key must be regenerated on the secondary site manually by running this command:

```
ALTER MASTER KEY
REGENERATE WITH ENCRYPTION
BY PASSWORD = '<password>';
```

<password> is the password that was used for creating the master key of the specific database during database installation.

Example:

ALTER MASTER KEY

REGENERATE WITH ENCRYPTION

BY PASSWORD = '4fdH5rr566!bv';

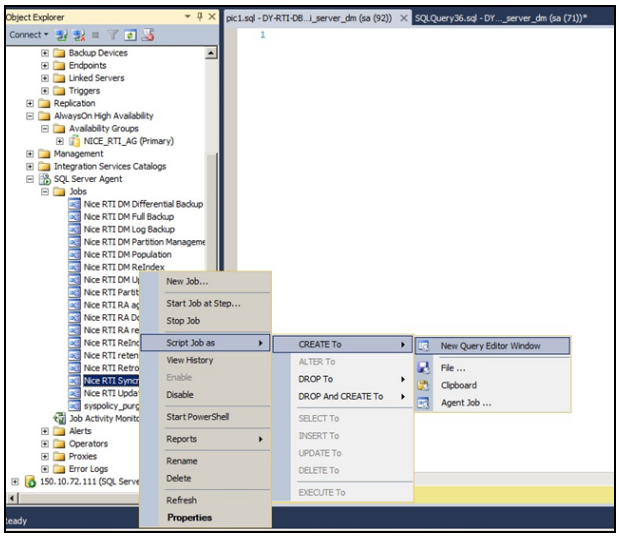
Post Installation Operations

After the RTI products have been installed, some manual installation operations should be performed on the secondary nodes:

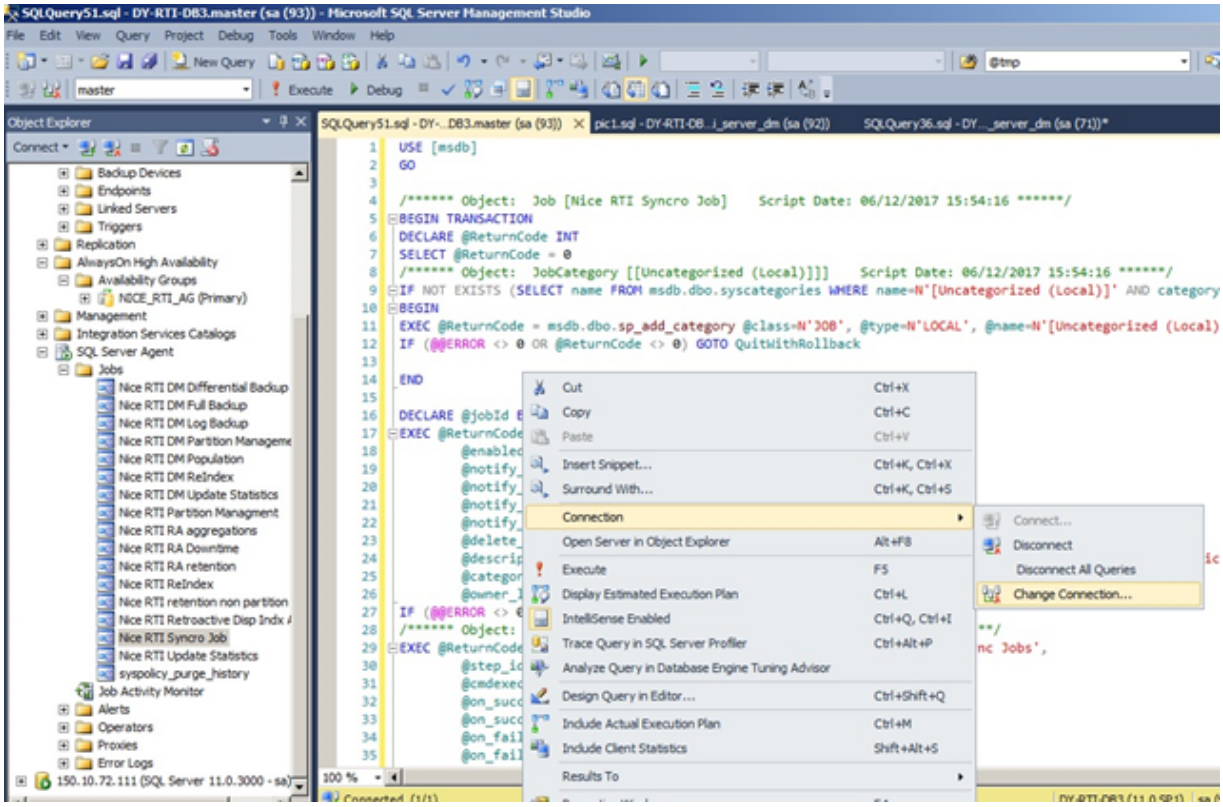
Installing NICE RTI Syncro Job on All Secondary Nodes

1. Connect to the primary node using SSMS.

2. In Object Explorer, expand the SQL Server Agent directory, and then expand the Jobs Directory.



3. Right-click on **Nice RTI Syncro Job** and select **Script Job As > Create to > New Query Editor Window**. The New Query Editor window opens, containing the code for the job creation.

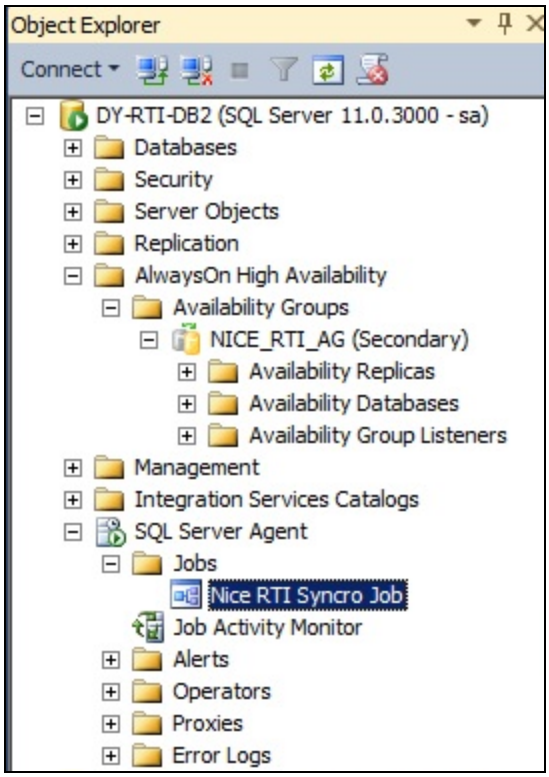


4. In the New Query Editor window, ensure the mouse cursor is on this window, then right-click and select **Connection > Change Connection**.

5. In the Connect to Database Engine window, enter the correct details and click **Connect** to connect to the secondary node.

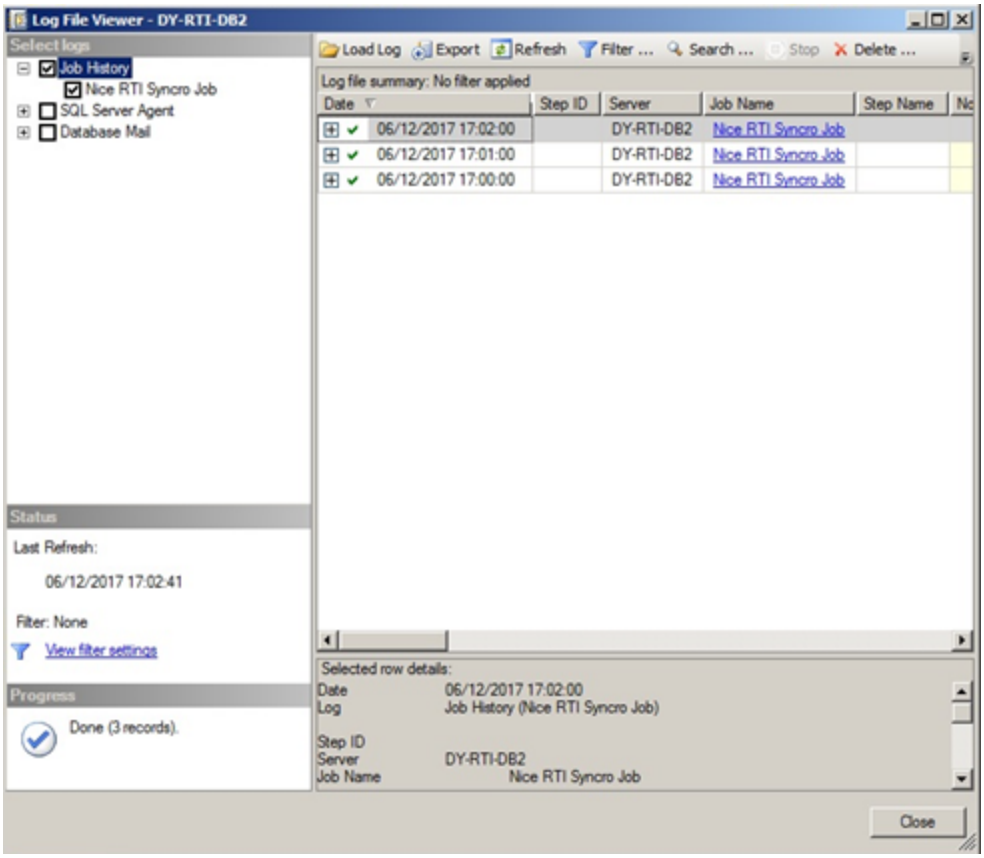
6. Run the script.

7. To verify that the job was created, in the Object Explorer pane, expand the SQL Server Agent directory, expand the Jobs directory and see if Nice RTI Syncro Job exists.



The Job is scheduled to run every minute, so after at least one minute, you need to check if the job has run:

- Stand on the job, right-click and select **View History**.
- In the Log File Viewer window, make sure the job ran successfully

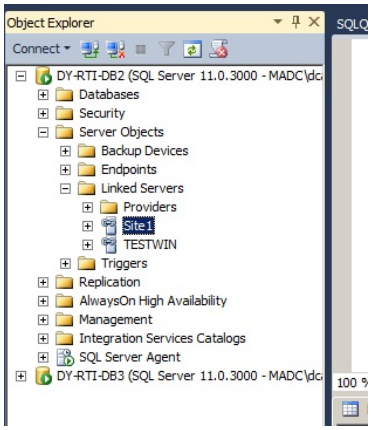


Creating a Linked server on the Secondary Node with the DataMart

This procedure relevant only for the nodes that host the Datamart database.

1. In SSMS, connect to the primary replica's SQL Server where the datamart DB exists.

2. In Object Explorer, expand **Server Objects > Linked Server** and mark the relevant linked server. Note that it will probably be named **Site n**, n being a number)



3. Right-click on the linked server and select **Script Linked Server as** > **Create to** > **New Query Editor Window**. A new query editor window opens, containing the script created for the linked server.

4. Right-click on the window and select **Connection** > **Change connection**.

5. In the Connect to Database Engine window, enter the correct details and click **Connect** to connect to the secondary node.

6. Run the script.

Copying the SSIS Package on the Secondary Node with the DataMart

As part of the post installation operations, you must manually copy the directory of the SSIS folder and all its content to an identical path on the secondary node with the Datamart. Make sure to assign the same permissions to the directory on the secondary node, as the permissions in the primary node.

Failover

Failover can occur in two scenarios:

- If a user initiates a failover
- Automatic failover can occur only if it was enabled when the AlwaysOn was configured, it meets its prerequisites, and the primary Replica is not available for any reason

During failover, the secondary replica assumes the role of the primary replica and its DB is set to read/write, instead of read only. The listener now redirects queries to the new primary replica. All connections to the DB are disconnected, and the DB is unavailable for a few seconds. It is highly recommended to configure the clients to reconnect when the connection to the DB is lost.

Failover on RTI System

The AlwaysOn, as designed by Microsoft replicates only the user databases, or in RTI terms only the operational database and the datamart database (if exists). It does not replicate the environment, i.e. jobs, backup devices and logins. In order to synchronize the environment to the secondary nodes, nice has defined the Nice RTI Syncro Job, which synchronizes those components.

If you run the RTI products on the same server of Engage, the environment synchronization is done by an Engage's job (Nice Synchro Job), and it does not replicate Logins.

Synchronizing Jobs

On the primary node, jobs that their name begins with "Nice RTI" are being documented in to a table in the user database, which as a part of the DB, it is replicated to the secondary node.

On the secondary node, there are two options: If this is the first time the job runs since a failover occurred, it recreates all the jobs documented in the table in the old primary node. If the job have already ran since the last failover, it only makes sure all Nice RTI related jobs are set to disable (except the Nice RTI Syncro Job).

To synchronize jobs:

1. Open SSMS and connect to the primary node.
2. In Object Explorer, expand **SQL Server Agent > Jobs**, and check that all RTI Nice's jobs are enabled.
3. In SSMS, connect to the secondary node.

4. If a failover have occurred in the past, in Object Explorer, expand **SQL Server Agent > Expand Jobs** and make sure all Nice RTI jobs are disabled, apart from the Synchro job. If a failover has never occurred (i.e. you've just installed the AlwaysOn) the Synchro Job is the only Nice RTI job you will see.

Also note that:

- Nice RTI Syncro Job must be created manually on the secondary node. For information look at Post installation Operations, Crate Nice Syncro Job on Secondary Node.
- The history of the jobs is not synchronized.
- Backup devices and logins are synchronized continually, every one minute, and failover does not affect their synchronization, unless they were changed less than a minute before they were changed.

Configuring DNS Alias Disaster Recovery

When setting up the DNS Alias/Public Load Balancer, you must:

- Import the additional certificates required for the DNS Alias/Load Balancer to all RT Designer and other client Virtual Machines (VMs).
- Configure OpenAM to enable clients to connect and authenticate through the DNS Alias/Load Balancer.

Creating and Importing Certificates for Disaster Recovery

Version 7.1 of APA only supports secure HTTPS communication and so security certificates for each server in the APA system must be imported to the client VMs in order for clients such as RT Designer to connect. With Disaster Recovery, additional certificates are required for the DNS Alias/Load Balancer.

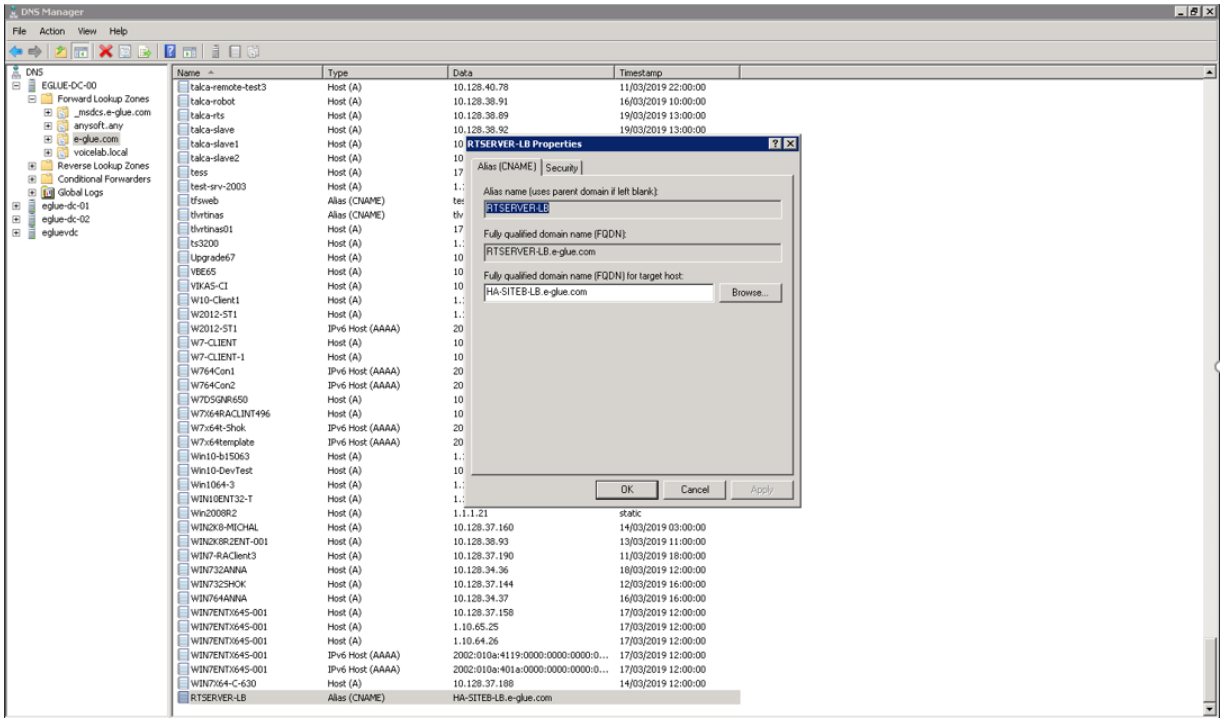
All certificates must be imported into Apache\certs and Tomcat\certs at the client VMs from both the active and passive systems. The certificates required are for the following:

- DNS Alias/Load Balancer
- Load Balancer for both systems
- All RT Servers for both systems

In particular for Disaster Recovery, you must create and import certificates for the DNS Alias/Load Balancer as described in the rest of this section.

To create and import the required certificates for the DNS Alias/Load Balancer:

1. Use the Windows DNS Manager to create a CNAME Alias for the load balancer and set its target to the load balancer for the active system. In the following example, the CNAME Alias is **RTSERVER-LB** and it points to the load balancer for the active system called **HA-SITEB-LB**.



2. Until you generate and import a security certificates for this new Alias, an error similar to the following occurs when clients such as RT Designer try to connect:

   ```
   2019-03-20 11:41:50,965 [1] ERROR Communication - ManualOpenAM.AcquireToken - Failed to login. Url: https://RTSERVER-LB.e-glue.com:1912/openam/json/authenticate, Exception:

   System.Net.WebException: The underlying connection was closed: Could not establish trust relationship for the SSL/TLS secure channel. ---> System.Security.Authentication.AuthenticationException: The remote certificate is invalid according to the validation procedure.
   ```

   You must generate a certificate for the new CNAME Alias at the load balancer for both the active and passive systems and import them into every client VM.

3. At the active system:

   a. Change directory to the Apache `conf` directory, for example `C:\nice_systems\RTServer\Apache\conf`.

   b. Open `httpd.conf` in a text editor and set ServerName to the CNAME Alias name. For example:

      ```
      ServerName RTSERVER-LB.e-glue.com
      ```

c. Add the Alias name to the Apache Web server by adding a ServerAlias entry in the `ncp.rts.conf` file. For example:

```
ServerAlias RTSERVER-LB.e-glue.com
```

For more details, see "Adding a Server Alias Name in the Apache Web Server" in the System Administration Guide.

d. Restart the RTServer Apache service of the load balancer.

e. Change directory to the Apache `bin` directory, for example `C:\nice_systems\RTServer\Apache\bin`.

f. Create a certificate for the Alias name with a CN of the Alias name. For example:

```
openssl.exe req -config "C:\nice_
systems\RTServer\Apache\conf\openssl.cnf" -nodes -x509 -days 8000 -subj
"/CN=RTSERVER-LB.e-glue.com" -newkey rsa:2048 -keyout "C:\nice_
systems\RTServer\Apache\certs\server.pem" -out "C:\nice_
systems\RTServer\Apache\certs\server.crt"
```

g. Copy the resulting `server.crt` and `server.pem` files to a temporary directory at one of the client VMs, for example to `C:\temp`.

h. Import `server.crt` into the trusted root certification authorities folder at the client VM.

i. Repeat steps g and h to import the certificate at all client VMs.

4. At the passive system, repeat steps 3a to 3i to import the passive certificate at all client VMs.

Configuring OpenAM for Disaster Recovery

To use a DNS alias for an installed Real-Time Server, some configuration fine tuning is required within the OpenAM configuration.

The web-policy-agent (WPA) configuration should be updated with the DNS alias entries, and the DNS alias should be added to the realm pool as well.

To configure OpenAM:

1. Add the CNAME Alias name created in the previous section to the realm properties using the OpenAM web console. For details, see "OpenAM - Adding a New Realm or DNS Alias" in the System Administration Guide.

2. Add the Alias name to the Apache Web server by adding a ServerAlias entry in the `ncp.rts.conf` file. For example:

```
ServerAlias RTSERVER-LB.e-glue.com
```

For more details, see "Adding a Server Alias Name in the Apache Web Server" in the System Administration Guide.

3. Update the WPA not-enforced URL list to add a new block of not enforced URLs that match the DNS alias record. For more details, see "OpenAM - Updating the WPA Not Enforced URL List" in the System Administration Guide.

4. Repeat these steps for the active and the passive system.

Setting Up DFS Replication Groups

SVN folders must be placed on a shared folder. The shared folder is replicated using the Microsoft DFS Replication method.

This method is an effective way to replicate data between servers across a room or on the other side of the world. DFS Replication uses remote differential compression (RDC) to replicate only the changes in a file on a block by block basis instead of replicating the entire file. Consequently, replication is very efficient even across limited bandwidth connections.

> **NOTE:** All servers must be on the same operating system.

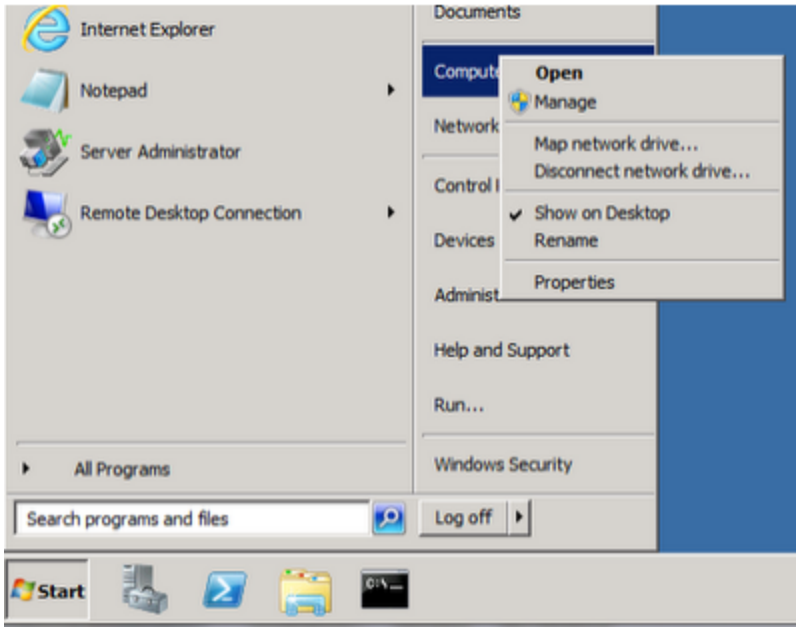
Setting Up DFS Replication in Windows Server 2008 R2

To set up SVN replication on Windows Server 2008 R2:

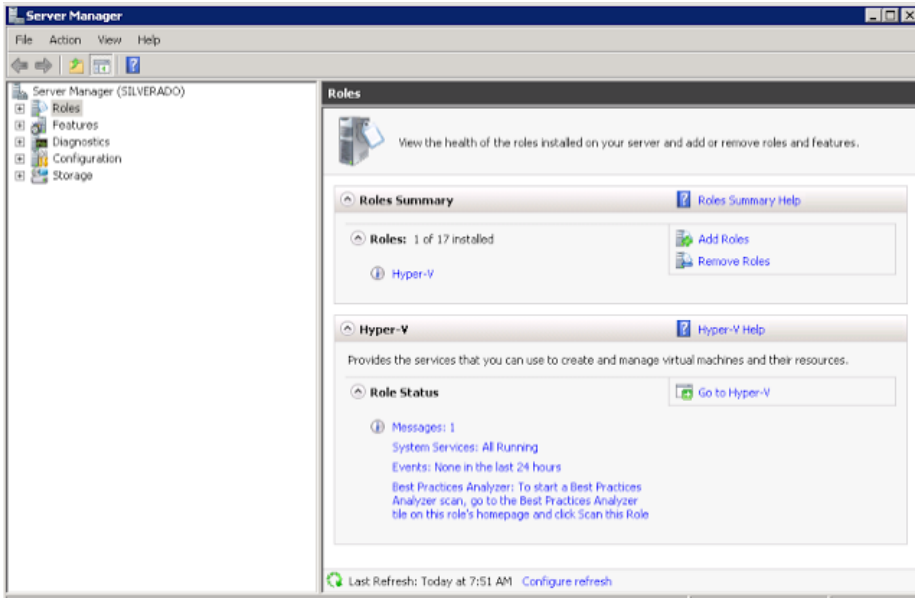
1. Add the **File Services** role.

   This installs the DFS service and management console. Repeat this step on each replication group member.

   a. From the **Start** menu, right-click **Computer**, and select **Manage**.

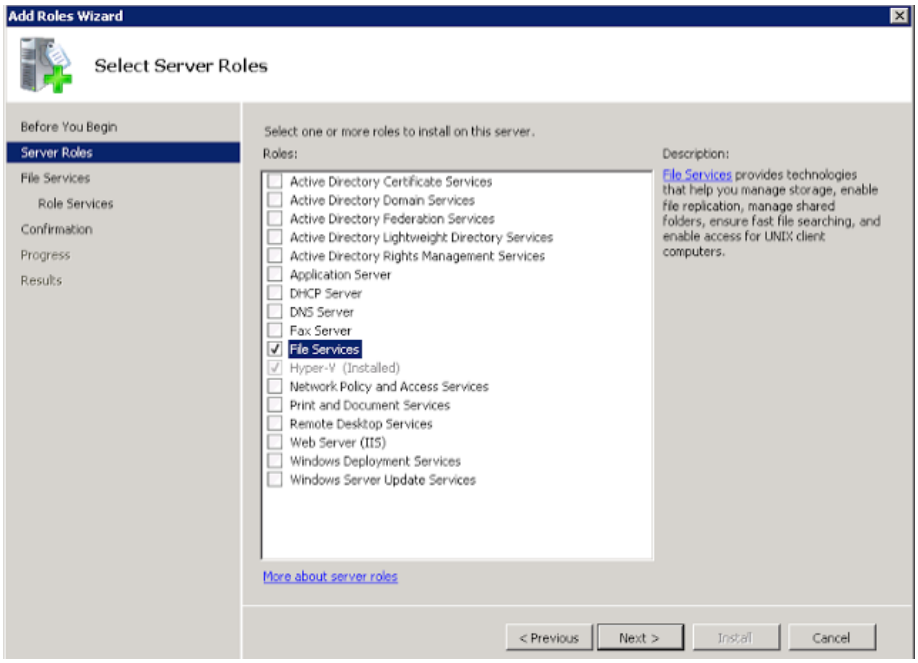


   b. In the **Server Manager**, Select **Roles** and in the **Roles Summary** pane, click **Add Roles**.



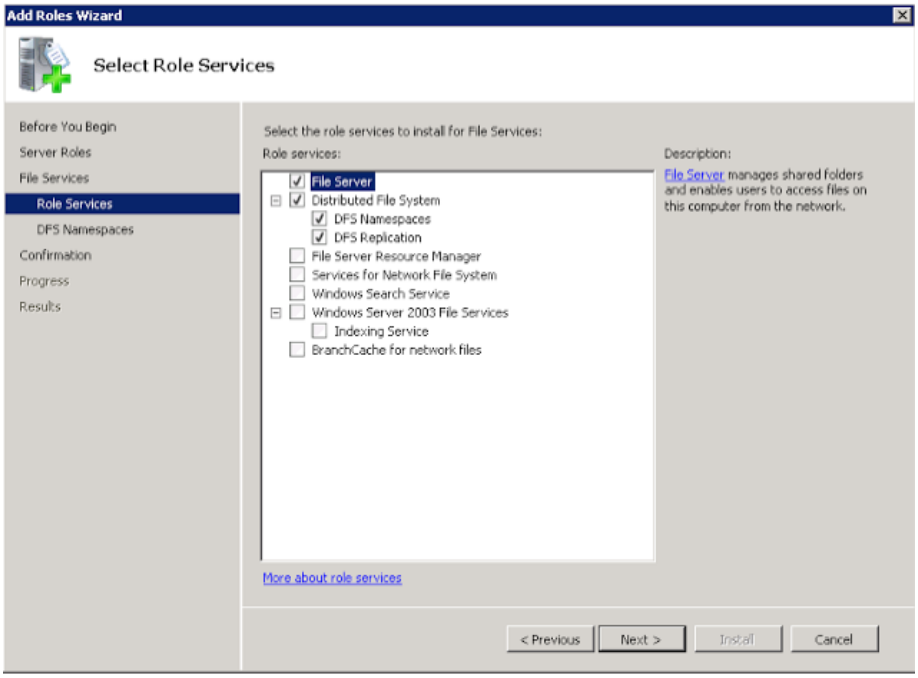
c. In the Before You Begin window, click **Next**.

d. In the **Select Server Roles** window, select the **File Services** role and click **Next**.

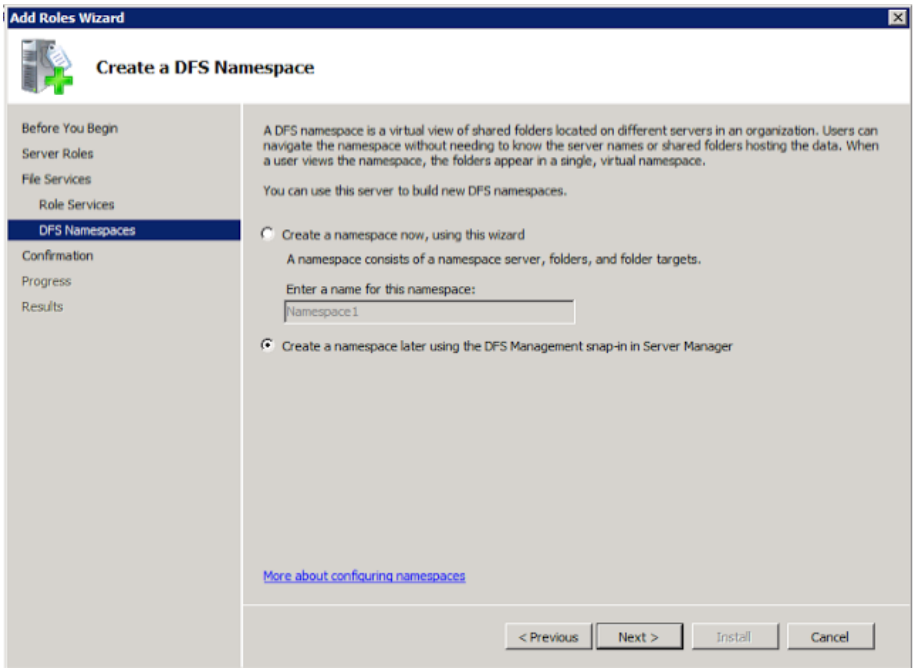


e. Click **Next** in the **File Services** window.

f. In the **Select Role Services** window, select the **Distributed File System** checkbox. Click **Next**.

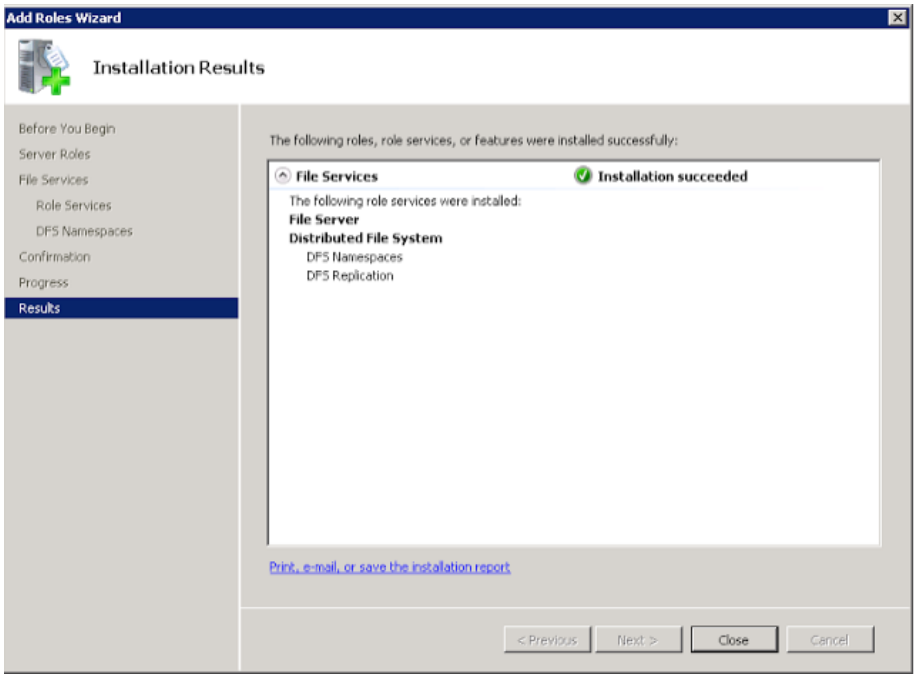


g. In the **Create a DFS Namespace** window, select the **Create a Namespace later using the DFS Management snap-in in Server Manager** checkbox. Click **Next**.

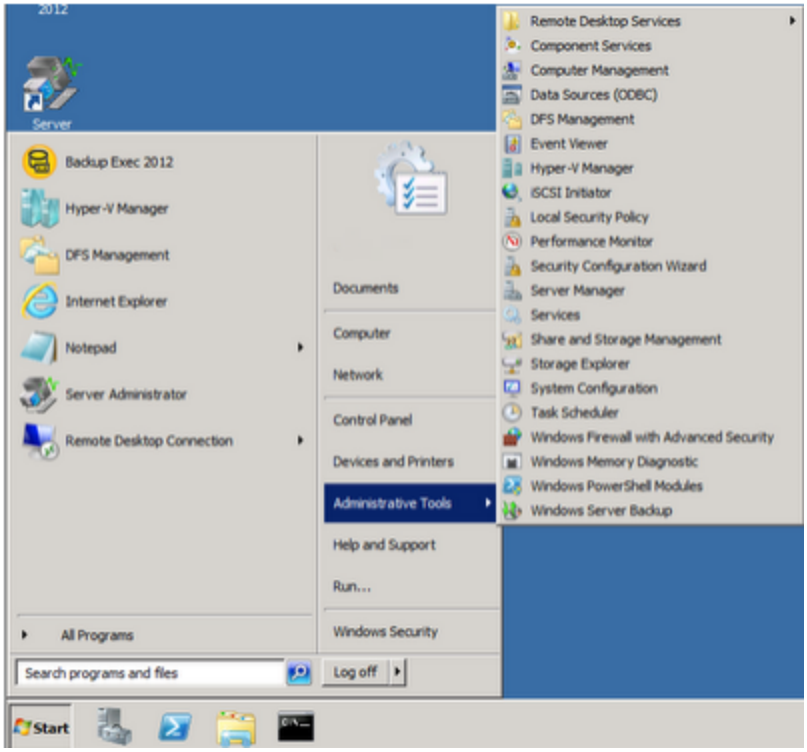


h. In the **Confirm Installation Selections** window, review the changes and click **Install**.

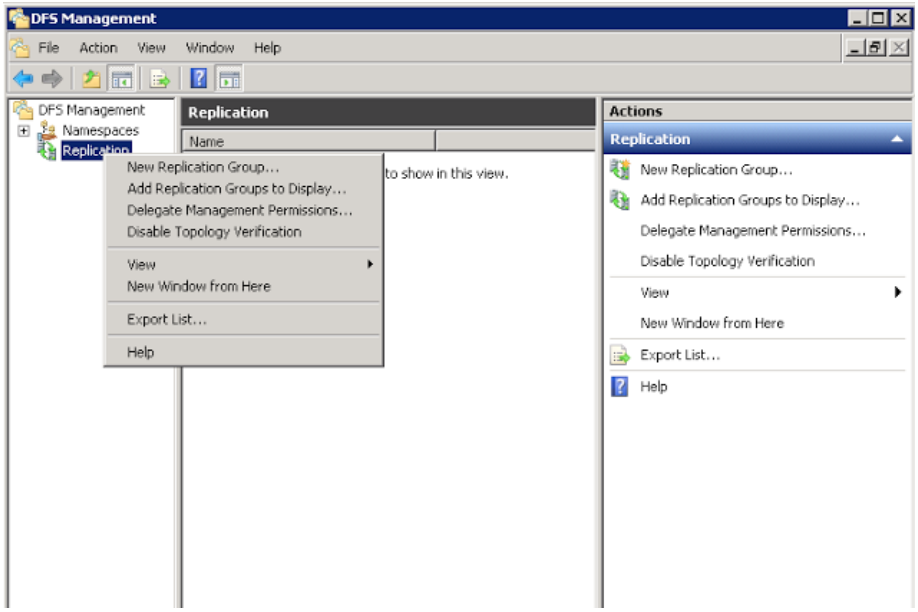
i. Review the **Installation Results** and click **Close** to end the installation wizard.



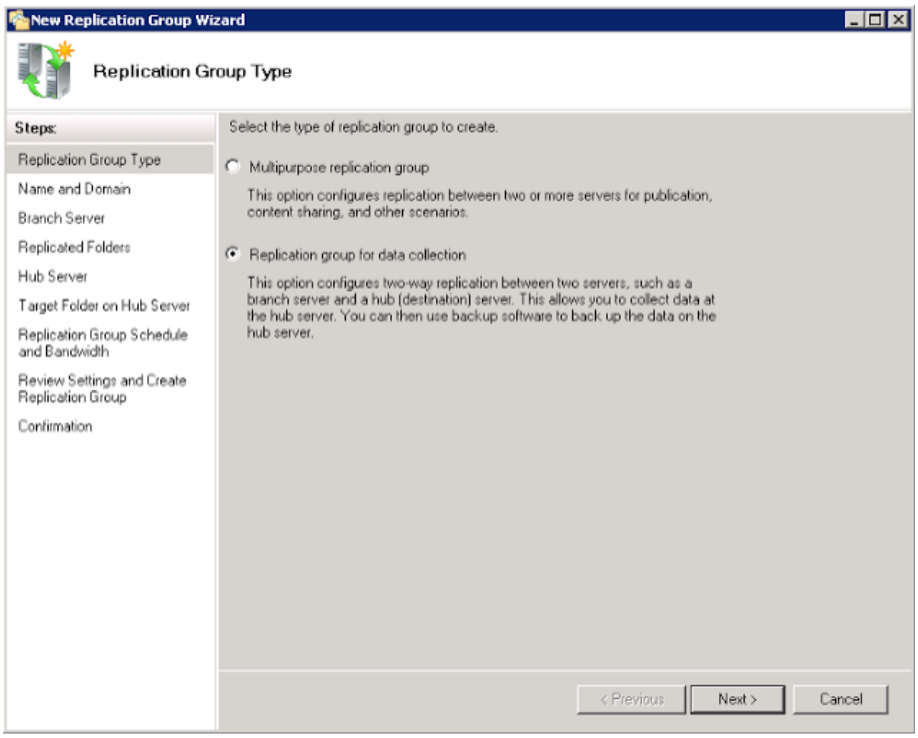
2. Set up replication on either the source or target server:
   a. From the Start menu, click **Administrative Tools** > **DFS Management**.



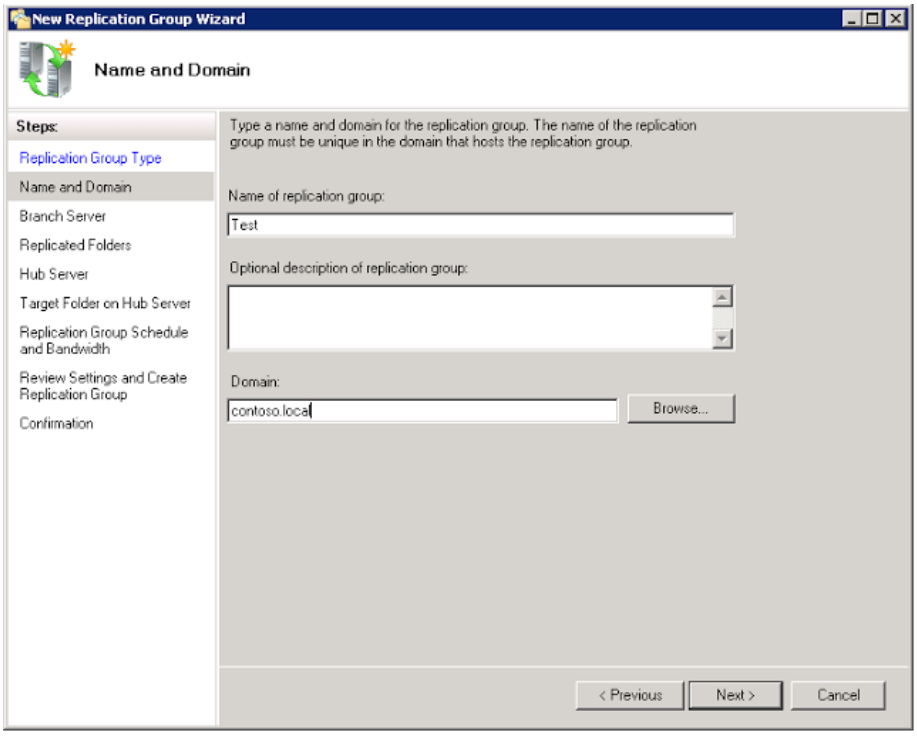
   b. In the **DFS Management** window, right-click **Replication**, and select **New Replication Group** to start the **New Replication Group Wizard**.



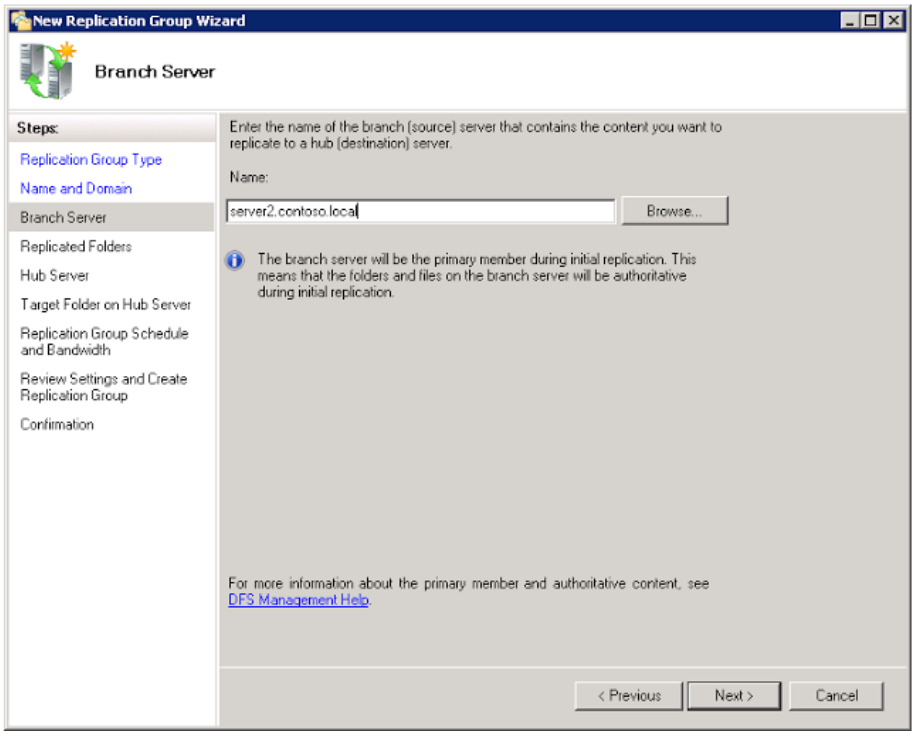
c. Select **Replication group for data collection** and click **Next**.



d. In the **Name of replication group dialog** field, enter a unique name for the replication group. In the **Domain** field, enter a domain that contains the replication group. Click **Next**.

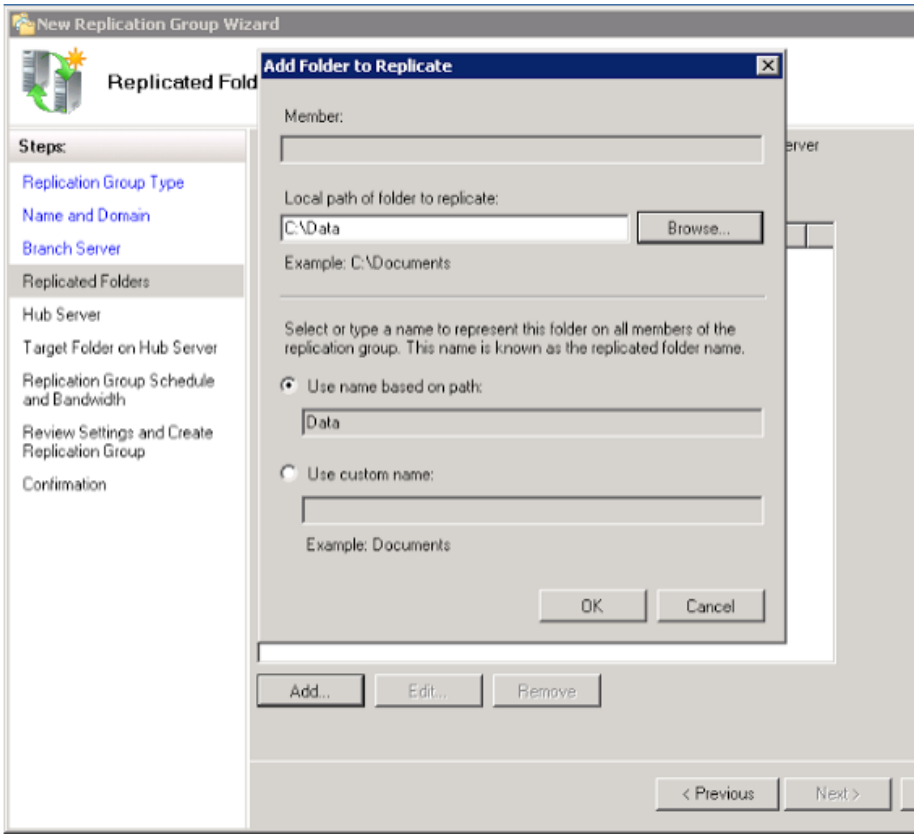


e. In the **Branch Server** window, enter the name of the source server in the **Name** field. All servers in a replication group must be in the same forest. Click **Next**.



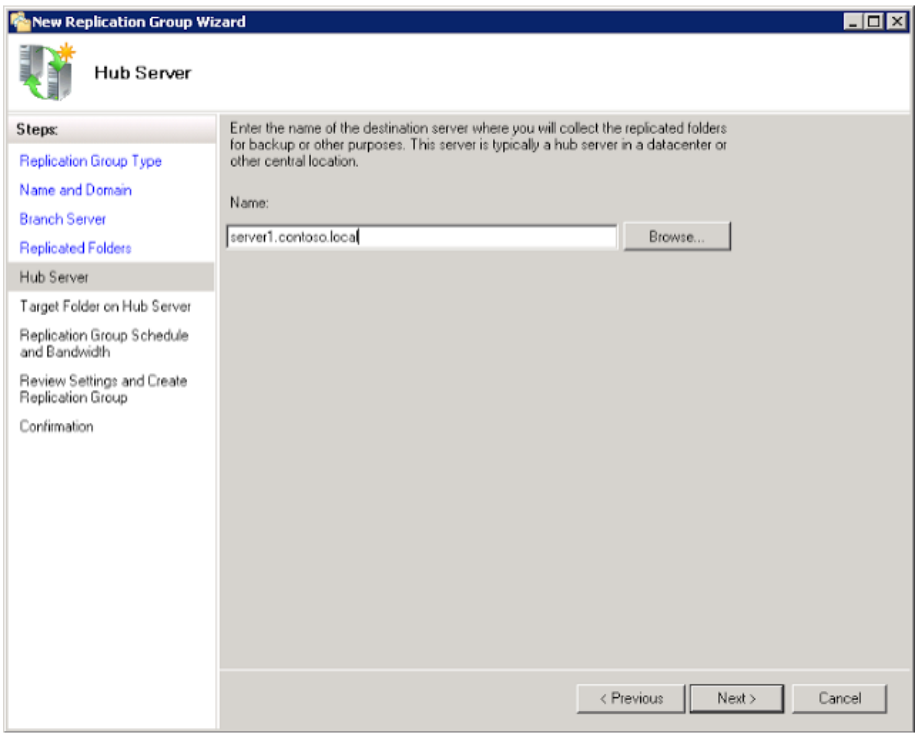
f. In the **Replicated Folders** window, click **Add** to specify which folders on the source servers to replicate.

g. In the **Local path of folder to replicate** field, enter the path or click **Browse** to browse to the folder. If you want a custom name for the replicated folder, click **Use custom name** and enter a new name. Click **OK** and repeat this step for each folder.

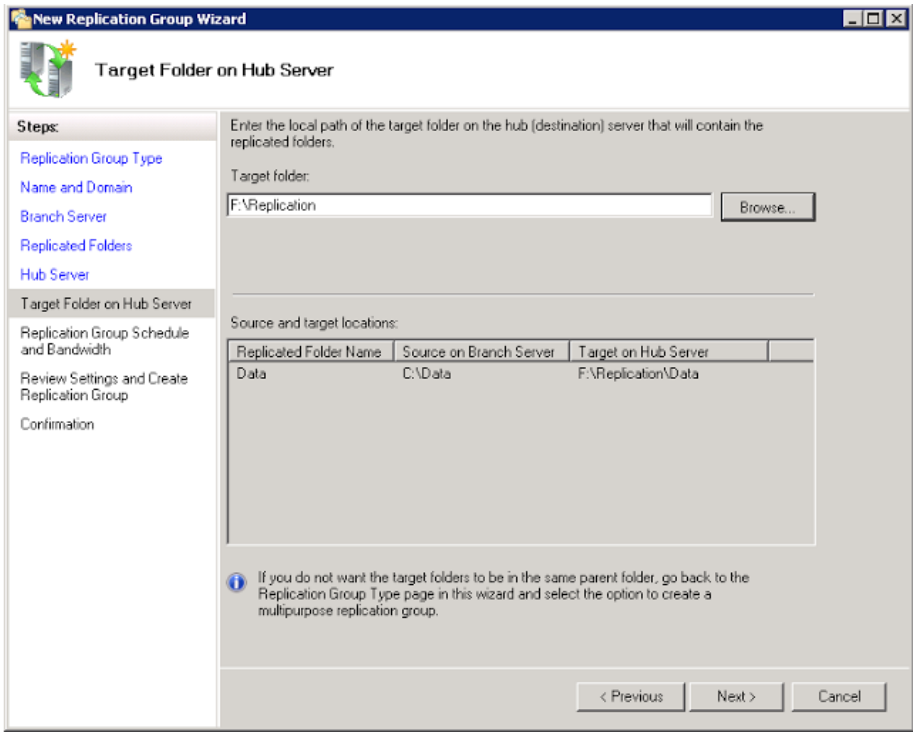


h. After adding all required folders for replication, click **Next**.

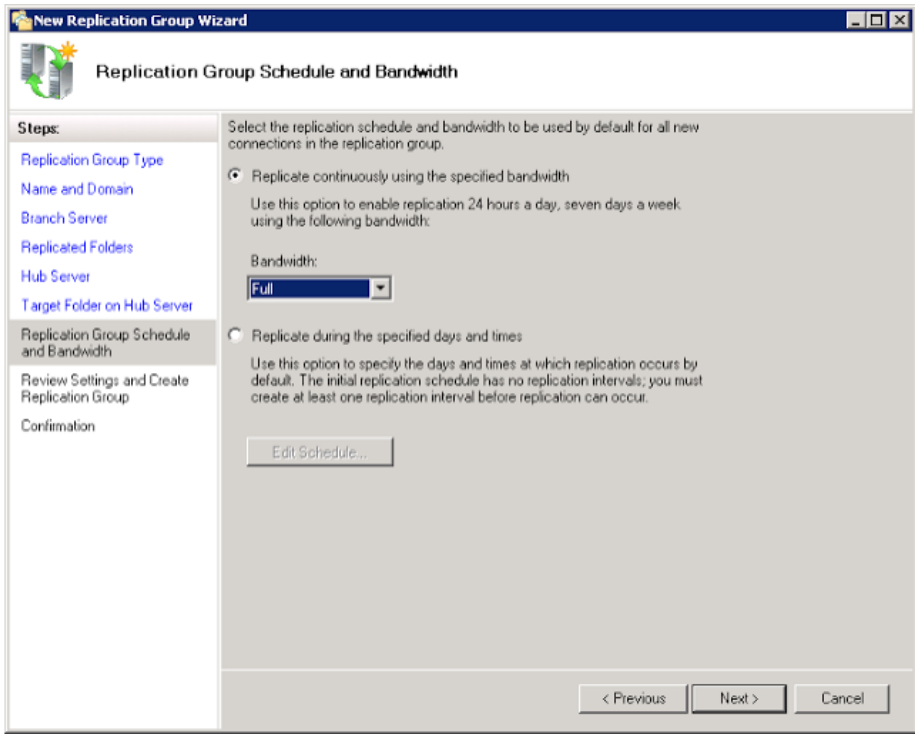
i. In the Hub Server window, enter the server name of the destination server in the **Name** field. Click **Next**.



j. In the **Target Folder on Hub Server** window, browse to the Target folder to which the source data is to be replicated and click **Next**.



k. Set the **Bandwidth** to be used to continuously replicate the data or select **Replicate during the specified days and times** to schedule the replication. Click **Next**.



l. Review the Replication group settings and click **Create** to build the replication group.

m. After the **New Replication Group Wizard** completes, click **Close**.

Setting Up DFS Replication in Windows Server 2012 R2 / 2016

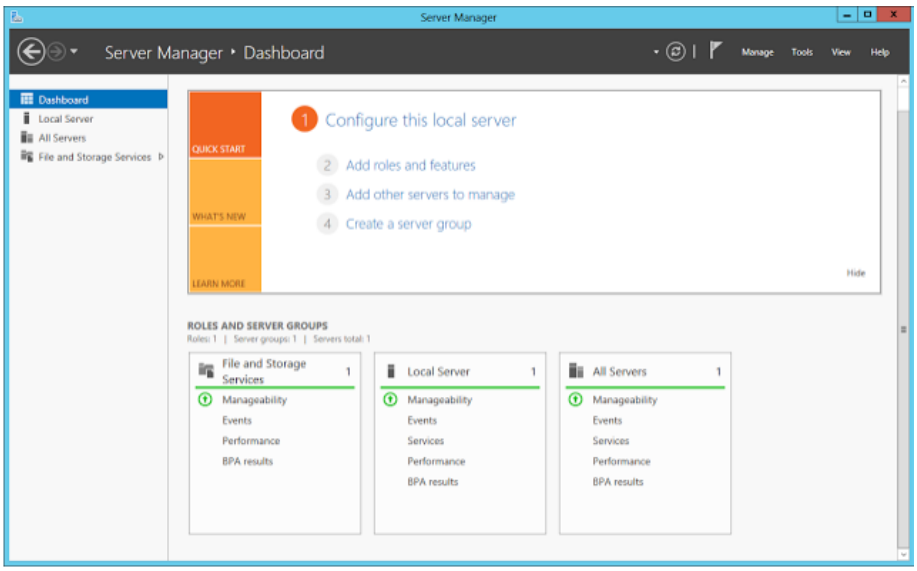
The following procedure includes two stages:

1. Installing the DFS Replication roles on each server participating in the replication group
2. Setting up replication between the servers

The steps are the same for Windows Server 2012 R2 and Windows Server 2016.

To install the DFS Replication role:

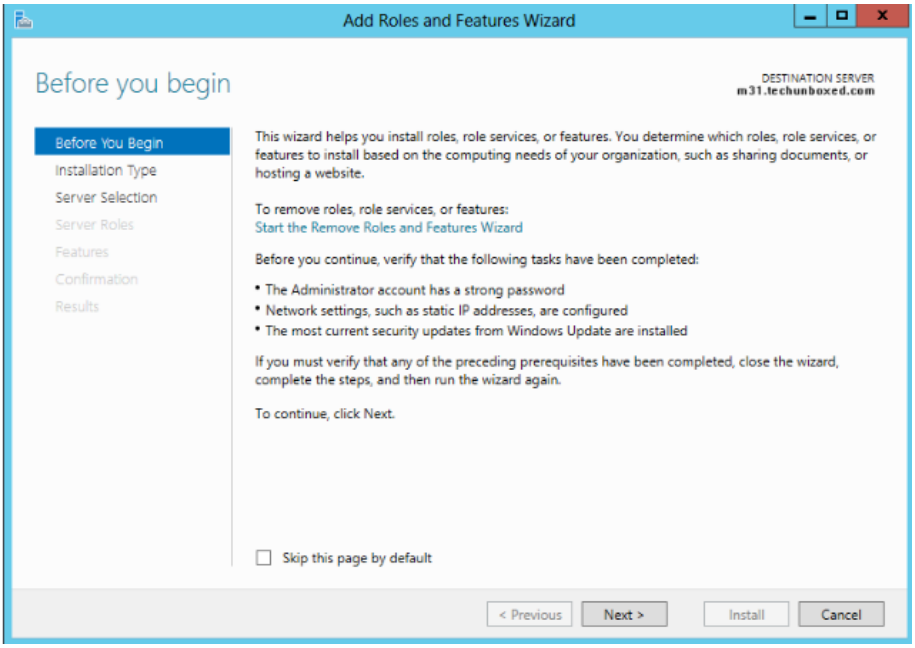
1. Open the **Server Manger** by clicking on the Server Manager icon on the task bar.



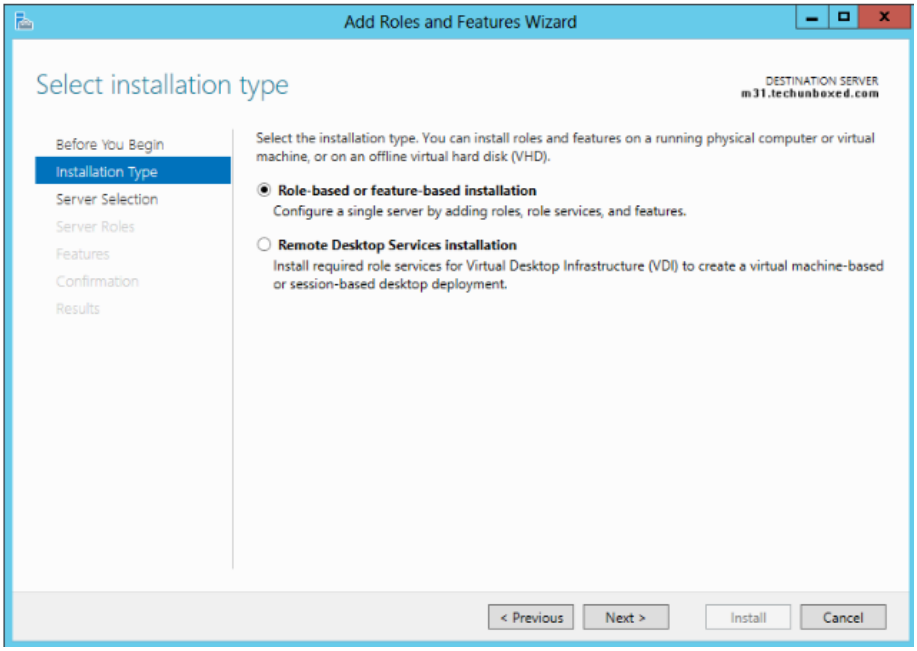
2. In the **Quick Start** section, click **Add roles and features**.

   NOTE: If there's no **Welcome** tile, it may be hidden. Click **View** on the menu bar, and click **Show Welcome Tile**.

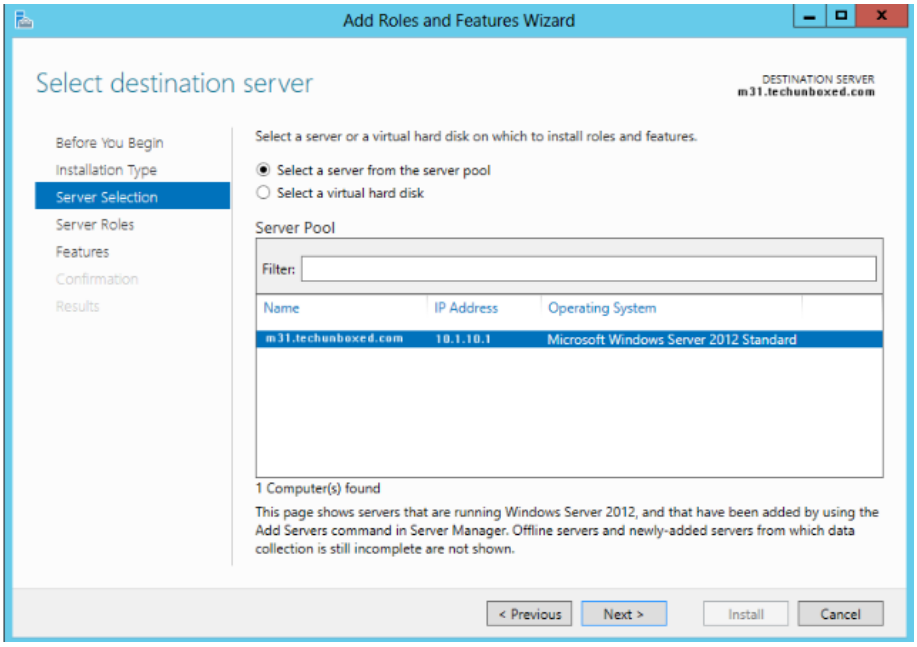
3. In the Before You Begin window, click **Next**.



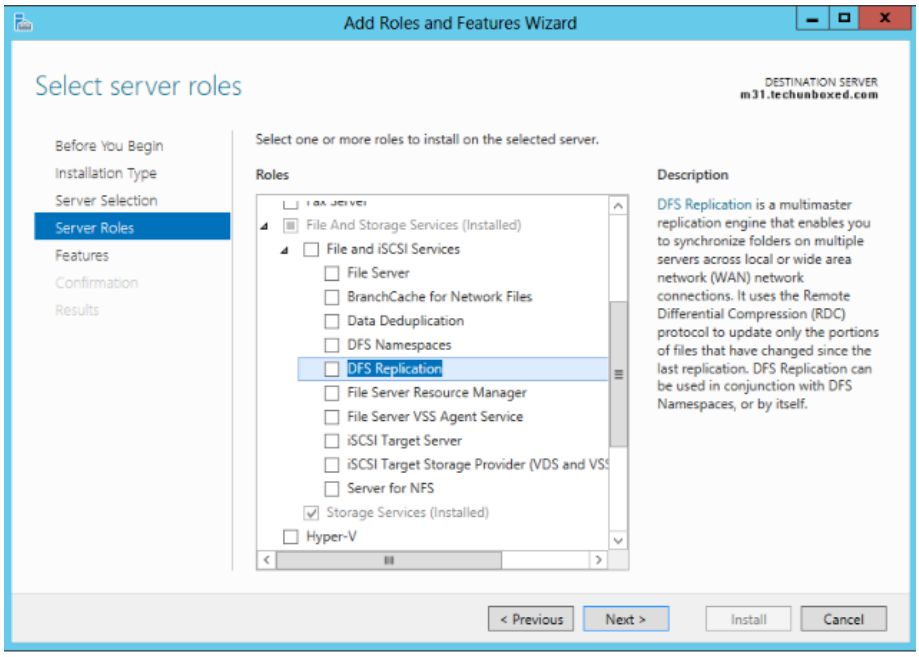
4. Select **Roll-based or feature-based installation** and click **Next**.



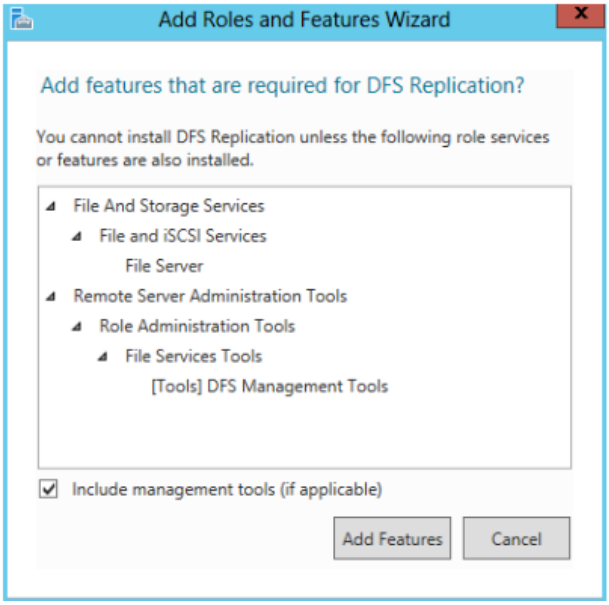
5. Select **Select a server from the server pool** and select the server on which you want to install DFS Replication. Click **Next**.



6. Under **Roles**, expand **File and Storage Services** > **File and iSCSI Services**, and select **DFS Replication**. Click **Next**.

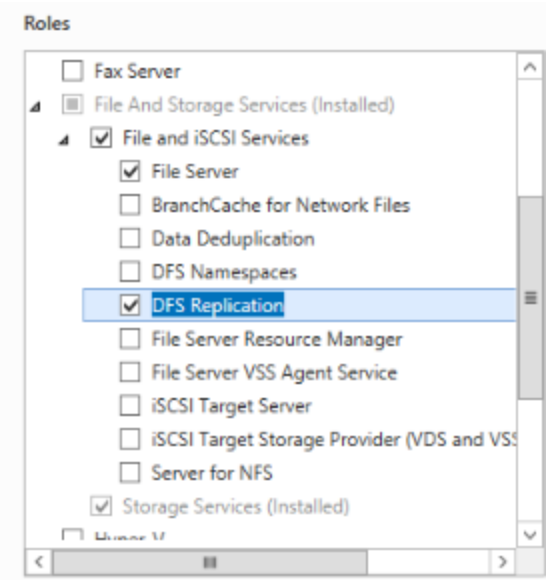


NOTE: If you have not already installed the features required for DFS Replication, the following message appears explaining which features and roles are installed along with DFS Replication.

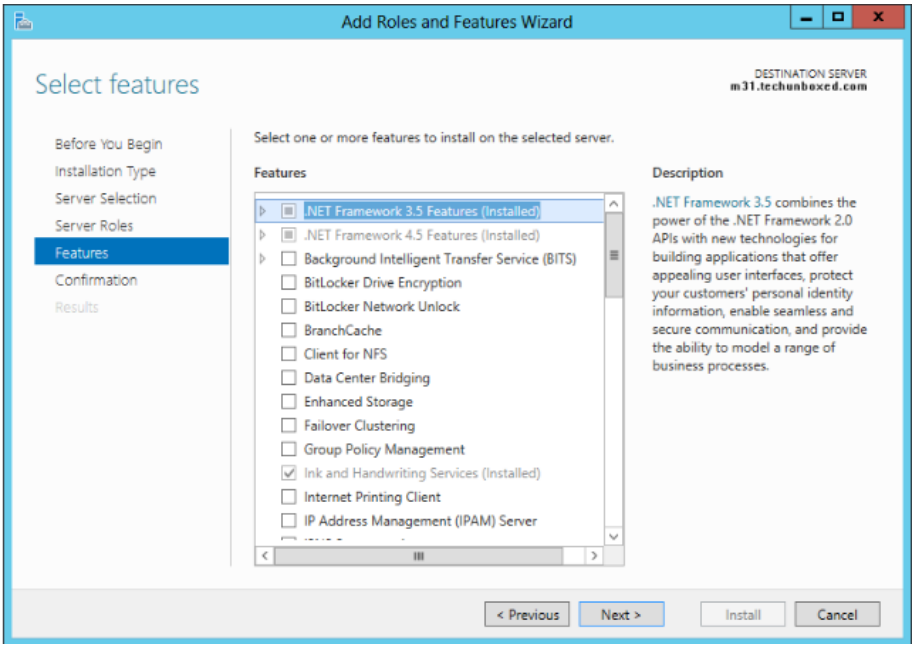


7. Click **Add Features**.

   Back to the **Select server roles** window. It should now show **DFS Replication** as selected along with the other roles required for DFS Replication. If everything looks ok, click **Next**.



8. The **Select features** window shows the features that will be added along with the DFS Replication role. Click **Next**.



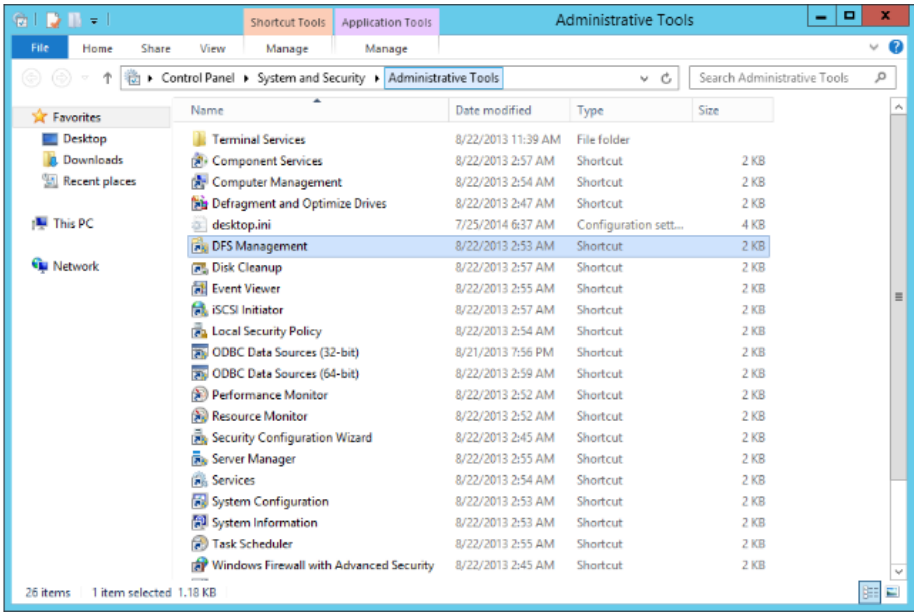
9. Review the installation selections and click **Install**.

10. Click **Close** when done.

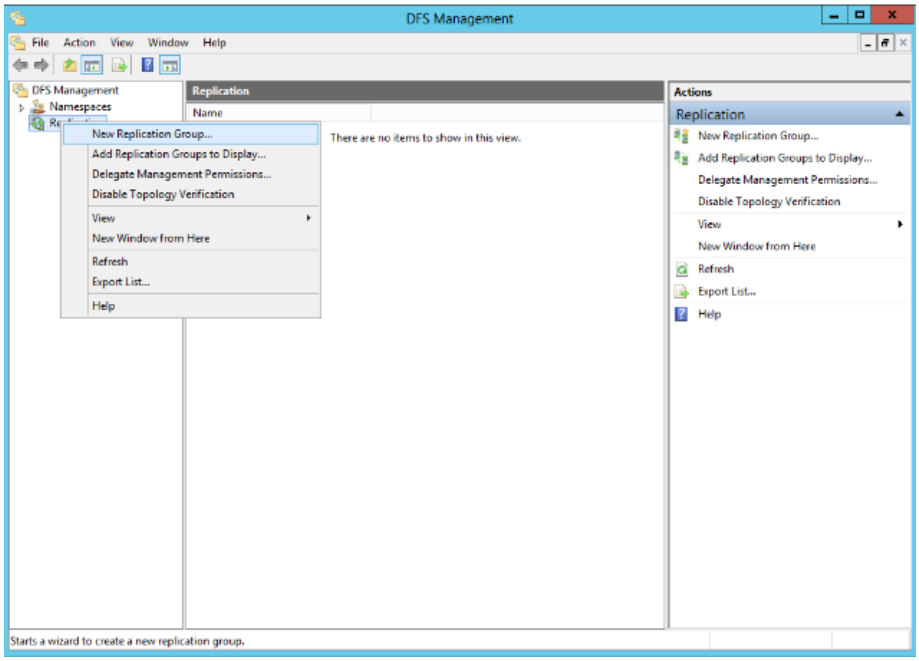
    Now that the DFS Replication role is installed, you can set up replication.

To configure replication between two servers:

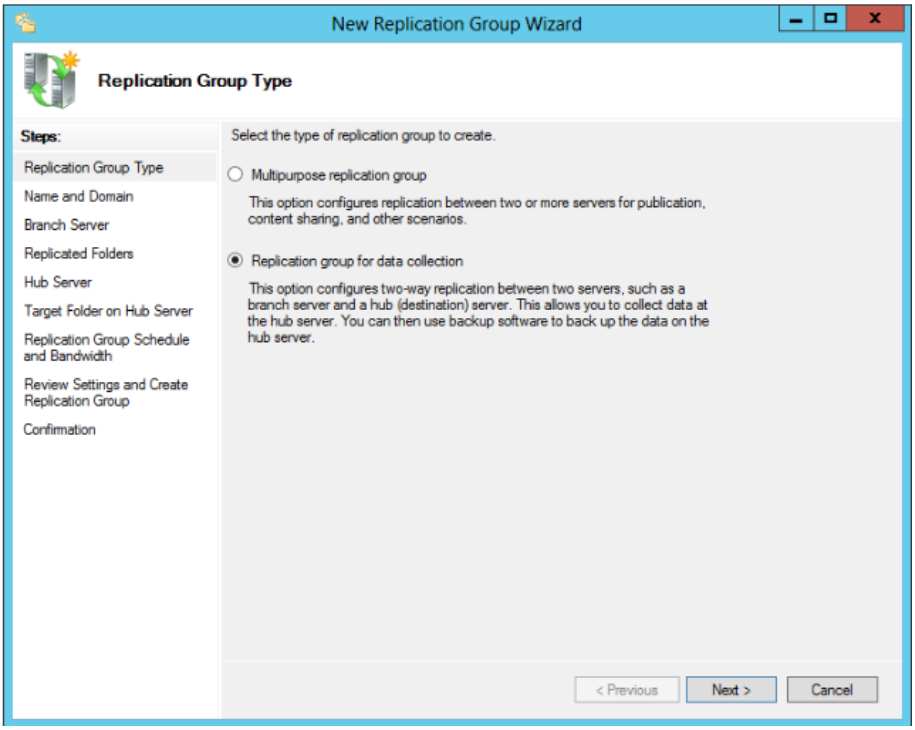
1. In the Control Panel, select **System and Security** > **Administrative Tools**. Then double-click **DFS Management**.



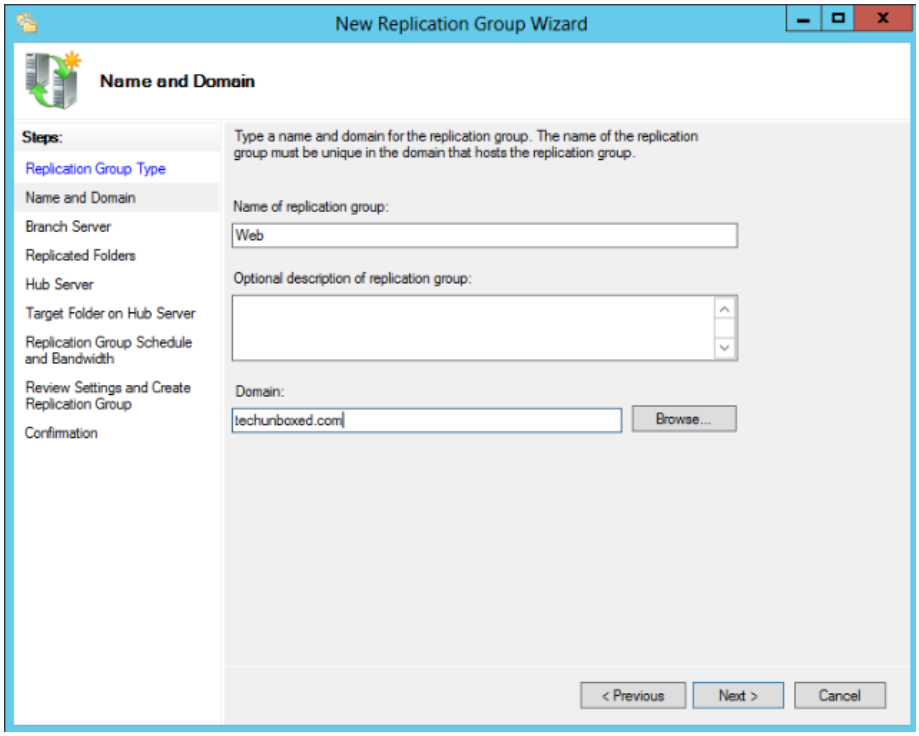
2. Right-click on **Replication** and click **New Replication Group...**



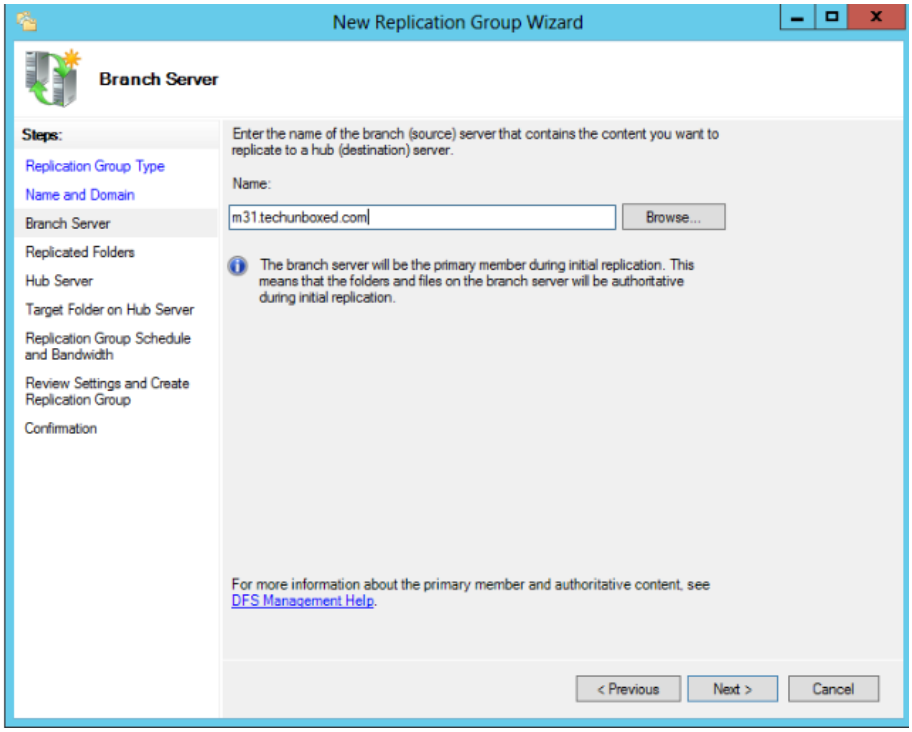
3. Select the **Replication group for data collection** option and click **Next**.



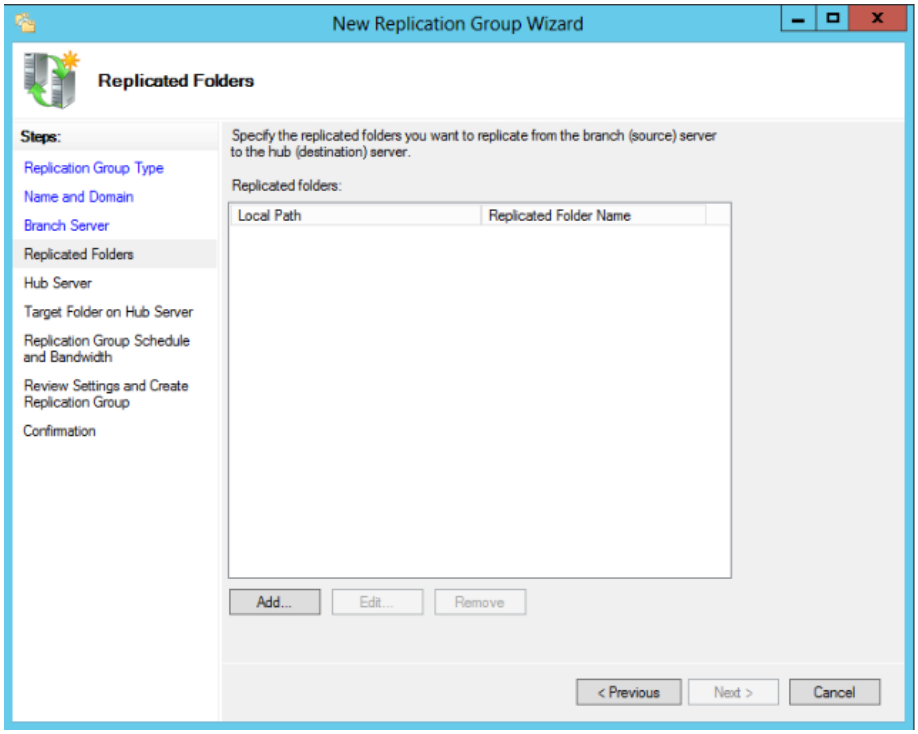
4. In the **Name of replication group** field, enter a descriptive and unique name . By default the **Domain** field contains the domain name of the server you're working with. Enter a different domain name if necessary.



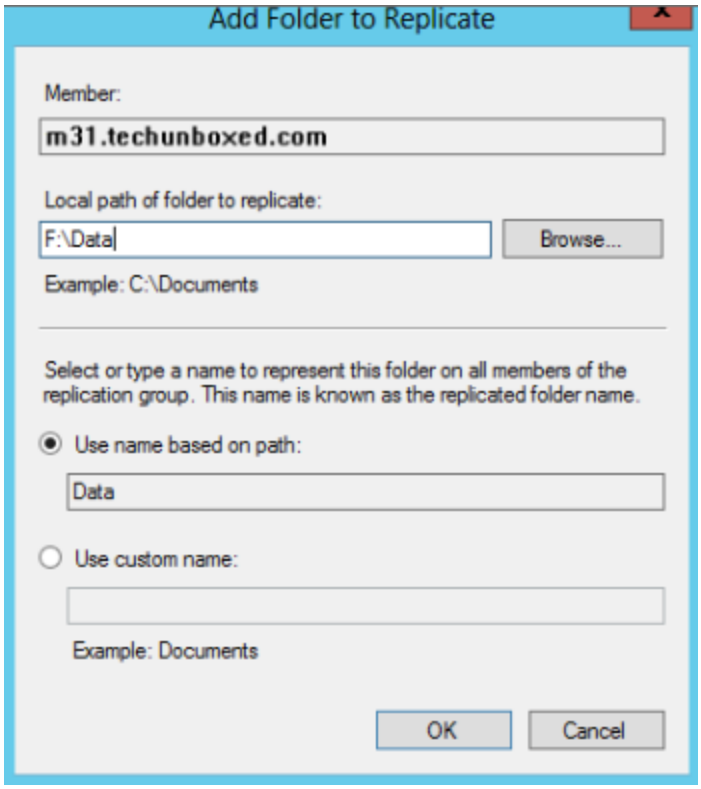
5. In the **Name** field, enter the name of the server containing the data that you want to replicate and click **Next**.



6. Click **Add** to define the folders that contain the data you want to replicate.

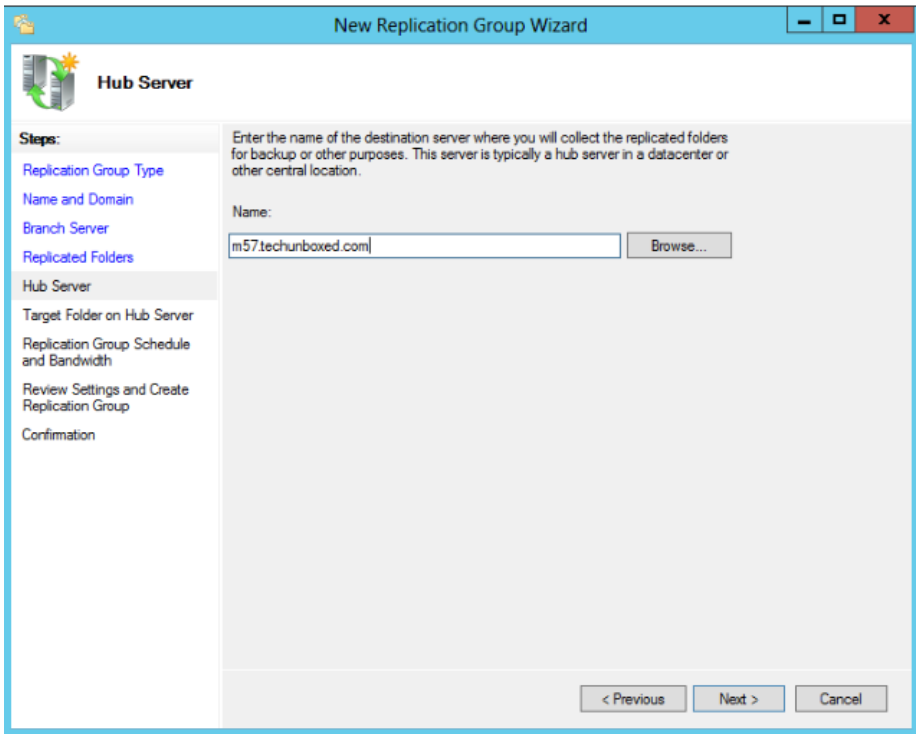


Enter or browse to the path of a folder to replicate in the **Local path of folder to replicate** field. You can enter a custom name to represent the folder or leave it set as the default. Click **OK**.

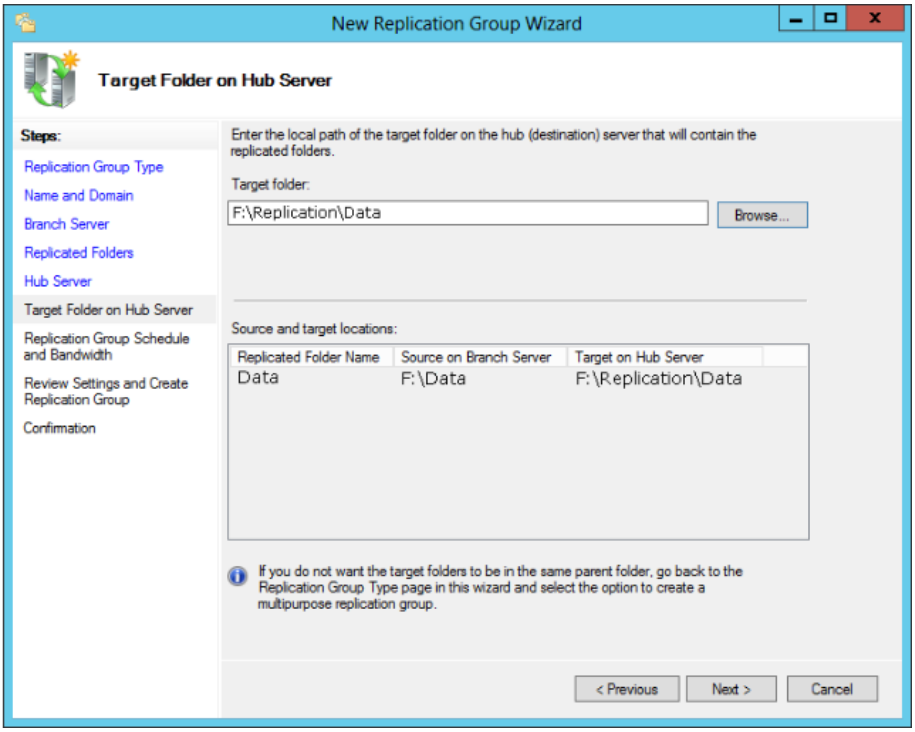


Click **Add** to add more folders to replicate (folders can be added later). When all the folders are added, click **Next**.

7. In the Hub Server window, enter the name of the server to be the target for the replicated data. Servers in replication groups must be in the same Active Directory domain. Click **Next**.



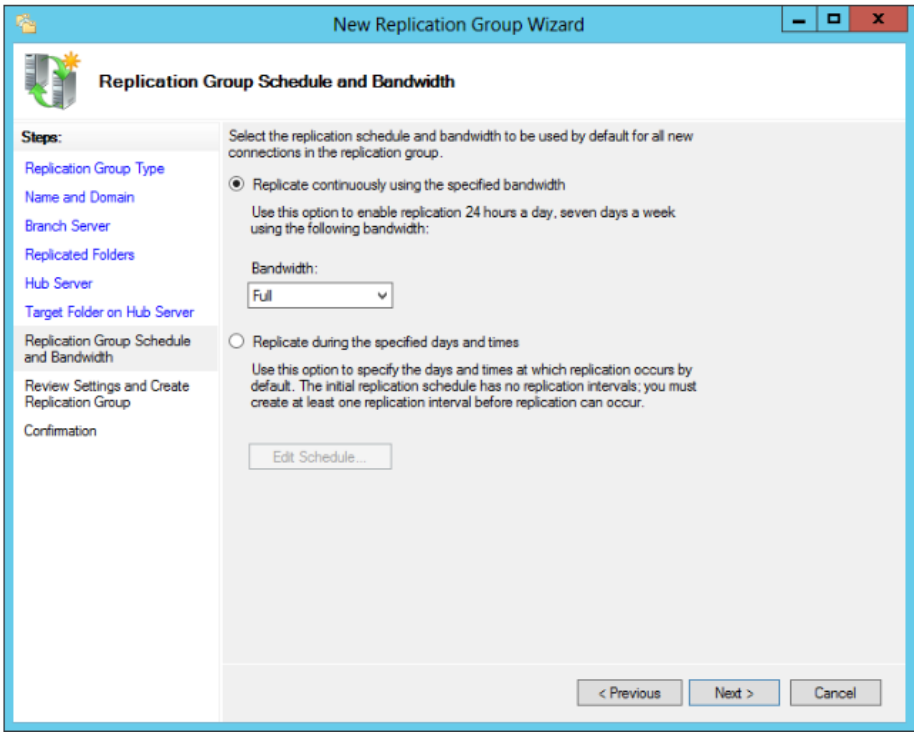
8. In the **Target folder** field, enter or browse to the path on the destination server where the replicated data is to reside. Verify the replication flow in the **Source and target locations** box. Click **Next**.



9. Select the bandwidth utilization method.

   Continuous replication takes place 24/7. You can select the amount of bandwidth for this option.

   The second method is to scheduled replication. Scheduled replication can be set to not replicate data during certain times and/or days of the week or at full or limited bandwidths. For example: Set a lower bandwidth during business hours, when network utilization is high, and full bandwidth at night and weekends.

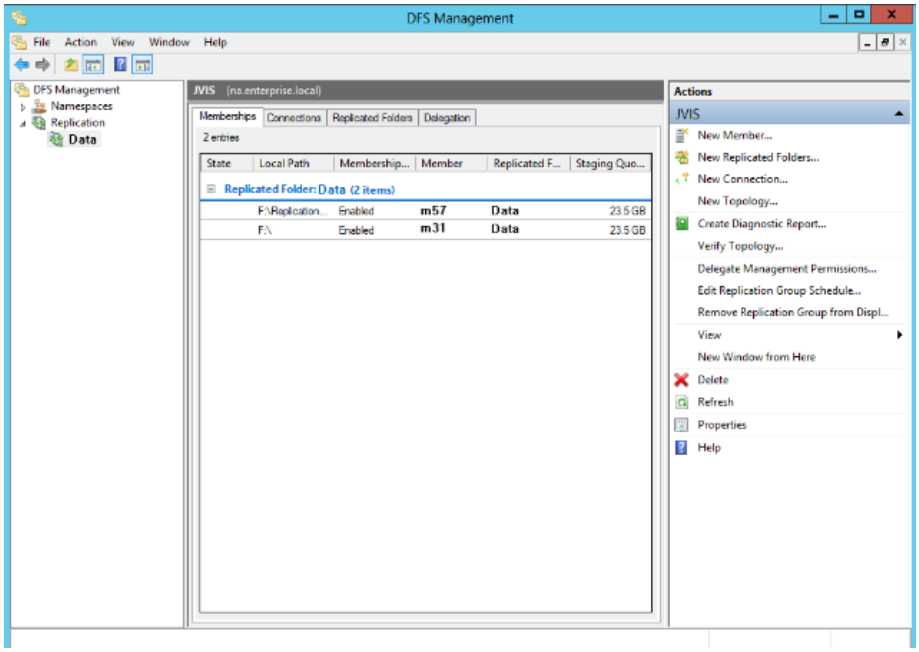


Once the replication bandwidth is set, click **Next**.

10. Review the replication group settings and click **Create**.

11. Read the message about replication delay and click **OK**.

12. Back in the DFS Management window, you can see the newly created replication group. Replication begins once the changes have been pushed to all the servers. Eventually, depending on bandwidth, etc., data starts showing up in the target folder on the destination server.


DFS Management
File Action View Window Help
DFS Management
Namespaces
Replication
Data
JVIS (na.enterprise.local)
Memberships Connections Replicated Folders Delegation
2 entries
State Local Path Membership... Member Replicated F... Staging Quo...
Replicated Folder: Data (2 items)
F:\Replication... Enabled m57 Data 23.5 GB
F:\ Enabled m31 Data 23.5 GB
Actions
JVIS
New Member...
New Replicated Folders...
New Connection...
New Topology...
Create Diagnostic Report...
Verify Topology...
Delegate Management Permissions...
Edit Replication Group Schedule...
Remove Replication Group from Displ...
View
New Window from Here
Delete
Refresh
Properties
Help

Configuring SVN Disaster Recovery

This section describes how to configure SVN for disaster recovery.

SVN folders must be placed on a shared folder. The shared folder is replicated using the Microsoft DFS Replication method. For more details, see Setting Up DFS Replication Groups on page 51.

Updating the Slave SVN

When a user publishes changes when connected to the one site, the Master SVN and Slave SVN of that site are both updated. However, when the data is replicated to the other Disaster Recovery site, only the Master SVN at that site is updated.

There are two possible solutions:

1. After failover from the active system to the passive system, you must run the four SVN scripts contained in the following folder on the Master SVN:

   `<RTServer Installation Path>\RTServer\Subversion`

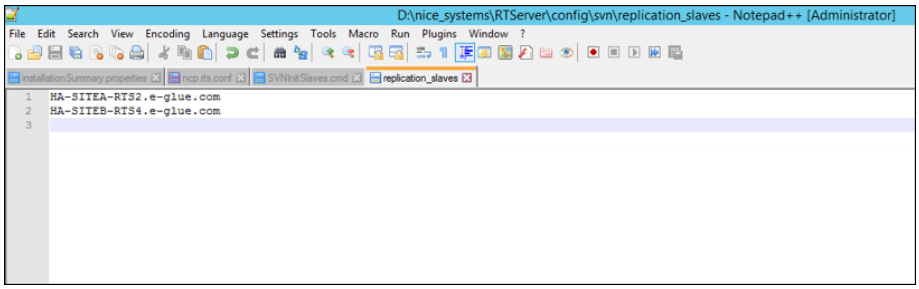
2. If both of the Disaster Recovery sites are active, you can force all slaves to be updated by editing SVN's `replication_slaves` file. You must edit the file at both systems to add the FQDN for the Slave SVNs of both systems. The file is located in the RTServer svn directory, for example `D:\nice_systems\RTServer\config\svn`.

   For example, if the FQDN of the Slave SVNs are:

   `HA-SITEB-RTS4.e-glue.com`

   `HA-SITEA-RTS2.e-glue.com`

   The file should contain the lines as shown in the following diagram:



Creating the SVN Service

This procedure describes recreating the SVN service.

To delete the service:

- Run the command: sc.exe delete "RTSERVER_Subversion"

To create a service:

- Create a new service pointing to the new drive.

  NOTE: Create the network drive letter using the cmd option as Administrator.

How to access a mapped directory from a windows service

If you have a problem connecting to the network drive, follow these guidelines:

NTFS symbolic links (symlinks) can refer to a UNC path but differ from shortcuts in that there is no redirect to the requested location. You can create a symlink like this:

mklink /D C:\myLink \\127.0.0.1\c$

When you open *C:\myLink*, the address of the folder you are in is *C:\myLink* and not *\\127.0.0.1\c$*, which is what you would get if myLink was a shortcut and not a symlink. This is significant if your application has compatibility issues with UNC paths.

Additionally the symlink exists in the file system and does not need to be recreated after logon as your average mapped drive requires (generally automated), and is therefore available to Windows services.

Configuring Elasticsearch Disaster Recovery

Back up the following folder and its contents on both Elasticsearch nodes:

`D:\nice_systems\RTServer\ElasticSearch\data`

Both nodes use different folders (node1\node3 and node2\node4) for storage on the Microsoft DFS system.

in order to retain an update folder you must create a third storage node in the replicated group.

Example

Replication group node 1:

- Site A Elasticsearch folder
- Site B Elasticsearch folder
- Backup storage Elasticsearch folder

Replication group node 2:

- Site A Elasticsearch folder
- Site B Elasticsearch folder
- Backup storage Elasticsearch folder

The replication group should be updated online so that the group always contains the updated sources.

Configuring ActiveMQ Disaster Recovery

Back up the following folder and its contents on both ActiveMQ nodes:

`D:\nice_systems\RTServer\ActiveMQ\data\kahadb`

Both nodes use different folders (node1\node3 and node2\node4) for storage on the Microsoft DFS system (as in SVN) .

in order to retain an update folder you must create a third storage node in the replicated group.

Example

Replication group node 1:

- Site A ActiveMQ folder
- Site B ActiveMQ folder
- Backup storage ActiveMQ folder

Replication group node 2:

- Site A ActiveMQ folder
- Site B ActiveMQ folder
- Backup storage ActiveMQ folder

The replication group should be updated online so that the group always contains the updated sources.

Configuring Cognos Disaster Recovery

You can install more than one Content Manager to ensure failover, and you can install Content Manager in a separate location than other components to enhance performance.

The Content Manager computers must know the location of the content store, the location of other Content Manager components, and the database that is used for notification.

In a distributed installation, at least one of the computers where you install Content Manager must be configured, running and accessible before you configure other computers in your IBM® Cognos® environment. This ensures that the certificate authority service, which is installed with Content Manager, is available to issue certificates to other computers.

Your installation may include more than one Content Manager, each on a different computer. One Content Manager computer is active and one or more Content Manager computers are on standby.

Permissions

You can install using either root or non-root authority.

Also, IBM Cognos BI respects the file mode creation mask (umask) of the account running the installation program. This affects only the installation directories. It does not affect the file permissions within the directories. However, run-time generated files, such as logs, respect the mask. We recommend umask 022 on the installation directory.

Rules for Configuring

In an installation where you have more than one Content Manager components, or where Content Manager is located in a separate location, at least one of the one Content Manager must be configured, running and accessible before you configure other components in your environment. This ensures that the certificate authority service, which is installed with Content Manager, is available to issue certificates to other IBM Cognos computers.

For information about the sequence of the installation process for distributed components, see Installation Sequence for Server Components.

Rules for Active Content Manager

If you are installing multiple Content Manager components, the first Content Manager computer that you start becomes the default active Content Manager. You can designate another Content Manager computer as default active, using IBM Cognos Administration.

The standby Content Manager computers are for failover protection. If the active Content Manager computer is not available because of a software or hardware failure, a standby Content Manager computer becomes active and requests are directed to it.

When the active Content Manager fails, unsaved session data is lost. When another Content Manager becomes active, users may be prompted to log on.

For information about activating a Content Manager service, see the Administration and Security Guide. For information about active and standby Content Manager components, see Active and Standby Content Manager Components.

In installations with multiple Content Managers, configure IBM Cognos BI to use an ISAPI gateway instead of the default CGI gateway. Otherwise, performance may be affected after failover.

Upgrading

If you are upgrading from ReportNet® or an earlier version of IBM Cognos BI, you can use the existing configuration data. However, some features in IBM Cognos BI are new and may require configuration.

PowerCubes

If you plan to install IBM Cognos Transformer and you will be using PowerCubes that are secured against an IBM Cognos Series 7 namespace, you must install Content Manager on a computer that supports IBM Cognos Series 7.

Active and Standby Content Manager Components

You can install any number of installations of Content Manager, although only one is active at any time. The other installations each act as a standby Content Manager.

The standby Content Manager components are for failover protection. If the active Content Manager is not available because of a software or hardware failure, a standby Content Manager becomes active and requests are directed to it.

When the active Content Manager fails, unsaved session data is lost. When another Content Manager becomes active, users may be prompted to log on.

By default, the first Content Manager installed with IBM® Cognos® BI is the active one. A IBM Cognos BI server administrator can change the default Content Manager and the active Content Manager at any time. When IBM Cognos BI is started, the default Content Manager locks the content store from access by all other installations of Content Manager. These other Content Manager installations enter standby mode.

This failover mechanism works because dispatchers and the active Content Manager routinely communicate with each other. If a dispatcher can no longer reach Content Manager, the dispatcher signals a standby Content Manager, which becomes the active Content Manager. The other installations of Content Manager remain in standby mode for continuing failover support. The standby Content Managers retrieve cryptographic settings, such as the common symmetric key (used to encrypt and decrypt data), from the active Content Manager.

Setting Up a Secured Clustered Database TLS 1.2 Environment

Use these procedures to set up a secured connection for a clustered database. This is generally used with disaster recovery.

1. Create a certificate for the database under the root certificate.

   See Securing the Database Servers in a Clustered Environment below

2. Import the root CA certificate, which now includes the database certificates, into the IBM Cognos key stores.

   See Securing the Cognos Server on page 87

3. Import the root CA certificate, which now includes the database certificates, into the Tomcat key stores.

   See Securing the Real-Time Server on page 87

Securing the Database Servers in a Clustered Environment

This procedure uses these servers as examples:

- AO listener: AOLSN.nice.com
- Database Node 1: DR-DBSERVER-A.nice.com
- Database Node 2: DR-DBSERVER-B.nice.com

NOTE: Highlighting indicates a variable that must be replaced.

The following is an example of how to request a SAN certificate. Your site's security IT team might require a modified version of this configuration to obtain a certificate with SAN.

➡ **On one of the RTServer nodes:**

1. Open a command prompt with elevated privileges.

2. Navigate to: *<RTServer Files>\RTServer\Apache\bin*

3. Execute this command:

   ```
   openssl.exe genrsa -des3 -out DB listener FQDN.key 2048
   ```

4. Follow the prompt and set a passphrase for the private security key.

5. Create a new file in notepad. Save the empty file to **C:** and name it **openssl SAN.cnf**.

6. Copy the following text into the new file with these changes:
   - Replace **AOLSN.nice.com** with the FQDN of the Listener.
   - Replace **DR-DBSERVER-A.nice.com** with the FQDN of one of the database nodes. Replace **DR-DBSERVER-B.nice.com** with the FQDN of a different database node, and so on.
   - Add or remove DNS lines as needed.

```
[ req ]
default_bits       = 2048
distinguished_name = req_distinguished_name
req_extensions     = req_ext
[ req_distinguished_name ]
countryName                 = Country Name (2 letter code)
stateOrProvinceName         = State or Province Name (full name)
localityName               = Locality Name (eg, city)
organizationName           = Organization Name (eg, company)
commonName                 = Common Name (e.g. server FQDN or YOUR
name)
[ req_ext ]
subjectAltName = @alt_names
[alt_names]
DNS.1   = AOLSN.nice.com
DNS.2   = DR-DBSERVER-A.nice.com
DNS.3   = DR-DBSERVER-B.nice.com
```

7. Execute this command:

```
openssl.exe req -new -key "server.key" -out "server.csr" -config
"C:\opensslSAN.cnf"
```

8. Follow the prompt and provide the relevant information:
   - (optional) Country name, State, Locality Name, Organization Name, and Organization Unit Name.
   - Common Name (the listener FQDN).
   - Do not provide an email address (leave blank).
   - A challenge password is not mandatory.

   The request file (*.csr) is created.

9. Submit the request file (*.csr) to your certificate issuer and you will get a ***.crt** or ***.cer** file.

10. In the CMD window, navigate to: *<RTServer Files>\RTServer\Apache\bin*

11. Create the *pfx file: run this command:

```
openssl.exe pkcs12 -keypbe PBE-SHA1-3DES -certpbe PBE-SHA1-3DES -export -in AOLSN.nice.com.cer -inkey AOLSN.nice.com.key -out AOLSN.nice.com.pfx -name "AOLSN.nice.com"
```

On every database node, force encryption:

1. Import the .pfx file into the database certificate store (use the Microsoft Managment Console (MMC)).

2. Give the user that starts the SQL Server service permissions to the .pfx file.

3. Open a CMD window and run this command to create the cert.txt file: `certutil -store my > cert.txt`

4. Open **cert.txt**, search for the .pfx and take The thumbprint can be located in the line that starts with "Cert Hash(sha1)"

5. Open the registry and Go to the selected DB service in HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\[*Instance ID]\MSQLServer\SuperSocketNetLib

6. Replace the certificate with the thumbprint \from the cert.txt file

7. In SQL Server Configuration Manager, you navigate to “SQL Server Network Configuration”, and then, for the SQL Server instance you want to enable SSL encryption, you right click on “Protocols for [instance name]” and enter its Properties.

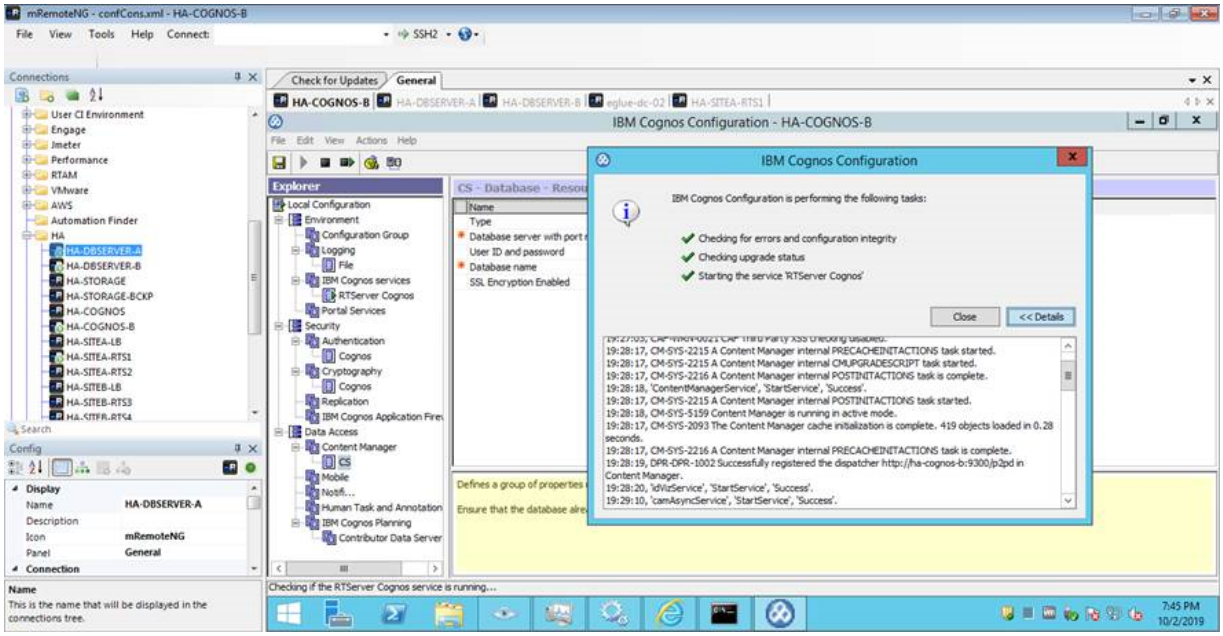
8. Restart the DB Server and validate that the DB service was started successfully on a secured node.

Related articles:

- How to Enable SSL Certificate-Based Encryption on a SQL Server Failover Cluster
- Configure SSL on a SQL Failover Cluster
- SQL Server fails to start with error 17182 "TDSSNIClient initialization failed with error 0xd, status code 0x38" when server is configured to use SSL

Securing the Cognos Server

Import the root CA certificate, which now includes the database certificates, into the IBM Cognos key stores.



Securing the Real-Time Server

Import the root CA certificate, which now includes the database certificates, into the Tomcat key stores. See *Importing Certificates for Secure Communication* in the *Real-Time Server Installation and Upgrade Guide*.

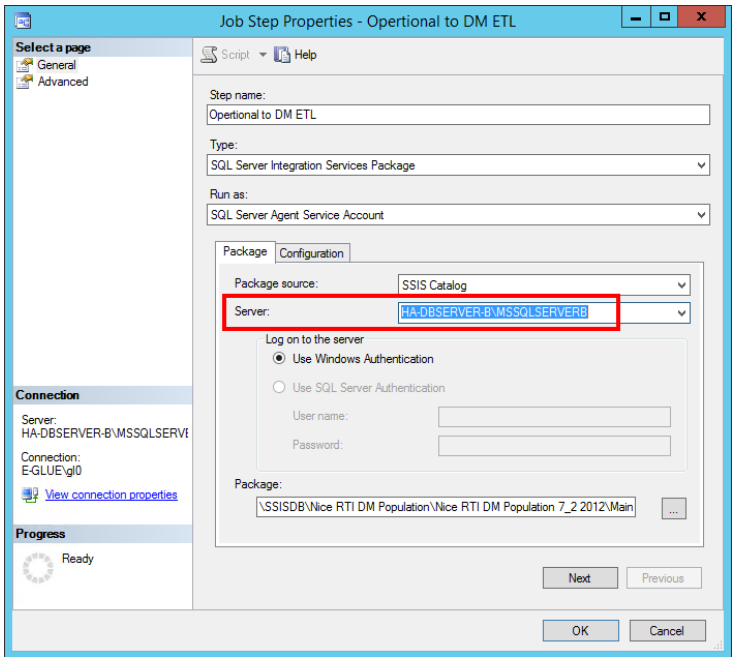
[This page intentionally left blank]

5

Performing a Disaster Recovery Failover Procedure

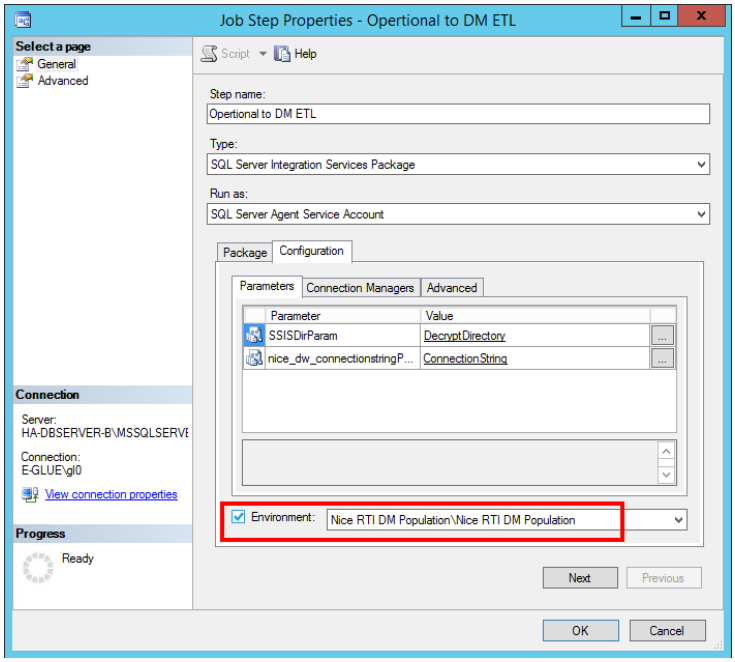
To perform a failover procedure, follow these steps to direct all services to the Disaster Recovery passive site.

1. If the SSISDB database is on a catalog: On the new Primary Real-Time server node, change the server name of the **Nice RTI DM Population By Catalog** job to the local server (current primary):
   a. Right-click **Nice RTI DM Population By Catalog** > **Properties** > **Steps** > **Edit**.



   b. On the **Package** tab update the **Server** name.

   c. On the **Configuration** tab select **Environment**.



2. On the **Operational** database, **CCM** database, and **SSISDB** database (if one exists) run these 2 commands:

```
open master key decryption by password = '<password>'
Alter Master Key Add encryption by Service Master Key
```

Validate the Environment

Check the following to validate the environment is working as expected:

1. Databases are working
2. AlwaysOn is working
3. Storage servers are available and running
4. All services are running.
   a. If you are using an Apache load balancer, check the following is running at the load balancer:
      - Apache service

   b. At the Master node, check the following are running:
      - RTServer MicroService - Password Store
      - Apache service

- SVN service
- Elasticsearch service
- ActiveMQ service
- Apache Tomcat service
- All other microservices

c. At the Slave node, check the following are running:
- RTServer MicroService - Password Store
- Apache service
- Elasticsearch service
- ActiveMQ service
- Tomcat service
- All other microservices

Move to the Passive Site

1. Stop the services at the active site:
   a. If you are using an Apache load balancer, stop the following at the load balancer:
      - Apache service

   b. At the Master node of the active site, stop the following:
      - Tomcat service
      - ElasticSearch service
      - ActiveMQ service
      - SVN service
      - All other micro services

   c. At the Slave node of the active site, stop the following:
      - Tomcat service
      - Elasticsearch service
      - ActiveMQ service
      - All other micro services

2. Change the DNS Alias to point to the passive site that is to be started.

3. Start the services at the passive site.
   a. At the Load Balancer, start the following:
      - Apache service
   b. At the Master node of passive site, start the following:
      - SVN service
      - Elasticsearch service
      - ActiveMQ service
      - Tomcat service
      - All other micro services
   c. At the Slave node of the passive site, start the following:-
      - Elasticsearch service
      - ActiveMQ service
      - Tomcat service
      - All other micro services

6

Testing the Disaster Recovery Flow

This section describes a sample testing scenario of Disaster Recovery environment with SVN and Desktop Analytics.

The scenarios below assume 2 sites: **Site A** and **Site B**. Both are the same: with a master and slave, and SVN version 19.

Before testing Site B, make sure that you run **init** and **proxy SVN** scripts on Site B.

Scenario 1

1. In Site A, publish a robotic solution and assign it to a robot. Make sure that the solution is added in the operational tables (META_DEPLOYED_SOLUTION, etc).

2. In Site B, make sure that the solution is added in the operational tables (META_DEPLOYED_ SOLUTION, etc).

Scenario 2

1. On Site A, publish a Desktop Analytics solution and assign it to a client and start this client.

2. Validate that the RT Client gets the correct version of the solution and an entry is made in the Desktop Analytics tables (wfm_app_monitoring_event).

3. Start a RT Client that is connected to Site B.

4. Validate that the RT Client gets the correct version of the solution and an entry is made in the Desktop Analytics tables (wfm_app_monitoring_event).

Scenario 3

1. On Site A, create a robotic solution with the invoke function, where the invocations will be added in Elasticsearch (head), ActiveMQ, and DataBase (RA_invoker_request and RA_invoker_request_ details).

2. Assign this solution to the robotic client and start this client.

6: Testing the Disaster Recovery Flow

3. Validate that the RT Client gets the correct version of the solution and an entry is made in the Desktop Analytics tables (wfm_app_monitoring_event).

4. Consider 50 invocations on ActiveMQ and 50 rows in Elasticsearch.

5. The robot will connect and start invocations. At invocation 30 in ActiveMQ, stop all the RT Server services at Site A.

6. Start all the RT Server services in Site B. The robot should connect to Site B and start invocations from 31st invocation in the ActiveMQ.

7. The 30th invocation should appear as a technical error in Automation Portal, after the TTL is past.