

**Setup of AOAG**

Version 1.0

|  |  |
| --- | --- |
| **KCT Number** | KCT0003604 |
| **KCT Name** | Corporate Production – Build SQL Always-On cluster |
| **Document Version** | Version 1.0 |
| **Service Owner** | SQLOps |
| **Approval Method** | SQLOpEsc |
| **Last Updated By** | Yusuf Anis |
| **Last Updated On** | July 7th, 2014 |

Table of Contents

[Purpose 3](#_Toc362249523)

[Audience 3](#_Toc362249524)

[Objective 3](#_Toc362249525)

[Ticket Flow 3](#_Toc362249526)

[Technical Scope for Execution 3](#_Toc362249527)

[Scoping Details / Template 4](#_Toc362249528)

[Pre-Execution Steps for creating new AlwaysOn Availability Group 5](#_Toc362249529)

[Execution Steps 5](#_Toc362249530)

[Names & configuration settings used in this setup. 5](#_Toc362249531)

[Preparation Steps 5](#_Toc362249532)

[Availability Group Creation 7](#_Toc362249533)

[Adding a Listener to an AOAG 18](#_Toc362249534)

[Escalation Matrix 25](#_Toc362249535)

[Appendix 25](#_Toc362249536)

[Document Administration Page 25](#_Toc362249537)

# Purpose

<http://infoplus30/GetDocument.aspx?DocumentID=D12-LV>

# Audience

<http://infoplus30/GetDocument.aspx?DocumentID=D12-LV>

# Objective

<http://infoplus30/GetDocument.aspx?DocumentID=D12-LV>.

# Ticket Flow

<http://infoplus30/GetDocument.aspx?DocumentID=D12-KX>

# Technical Scope for Execution

**In Scope:**

This document has following scope of activities for the subject KCT.

1. Create a New SQL Server 2012 AlwaysOn Availability Group.

These changes will be **scheduled activity** (**SHOULD ONLY BE EXECUTED DURING PROVIDED DOWNTIME WINDOW ONLY. ANY DOUBT ON DOWNTIME WINDOW SHOULD GET CLARIFIED WITH REQUESTOR BEFORE EXECUTION**) specified in the RFC.

1. Configuring cluster properties of Availability Group resources.

**Out of Scope:**

Following activities are out of scope.

1. Adding Nodes to an existing Availability Group
2. Creating the Windows Failover Cluster
3. Synchronizing additional User Databases within an Availability Group

# Scoping Details / Template

1. Verify if servers in request are pinging and TS from local machine.
2. Use this template to capture details. All the requesters are expected to provide AlwaysOn requests using this template.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ticket Info** | | | | |
| RFC# |  | | | |
| TM# |  | | | |
| **Environment Verified** | | | | |
| RDP / SQL Permissions granted to PSITADM? | Yes | | No | |
| Service running under? | Local | | Domain | |
| Business justification if running under local account. |  | | | |
| Symmetric storage Checked? | Yes | | No | |
| **AOAG Details** | | | | |
| Configuration Type | **1** | **2** | **3** | **4** |
| Complete Instance Name(s) |  | | | |
| Availability Group Name |  | | | |
| Listener Name |  | | | |
| IP address(es) |  | | | |
| For each Replica choose one (async for multi-subnet) | Async / Sync | | | |
| Failover type (AUTO only if Sync & not a FCI) | Manual / Auto | | | |
| Sync & Failover Options | Async / Sync | | Manual / Auto | |
| Primary |  | |  | |
| Replica 1 |  | |  | |
| Replica 2 (if needed) |  | |  | |
| Replica 3 (if needed) |  | |  | |
| Replica 4 (if needed) |  | |  | |

**Possible Blockers**

Following could be possible blockers while implementing this change activity.

1. Nodes which are in request are not able to ping or TS.
2. Nodes are not yet part of a Windows Failover Cluster.
3. Listener IP / DNS pinging from local machine.
4. Listener IP not from the correct subnet.
5. \*Nodes do not have symmetric storage.
6. \*SQL Services are running on managed accounts.
7. #Cluster Account does not have object creation right in the AD or the Resource Names have not been pre-staged.

Contact requestor to have correct information for any of these listed blockers.

# Pre-Execution Steps for creating new AlwaysOn Availability Group

1. Verify the possible blockers.
2. Windows patches can be installed before or after this configuration.
3. SQL Server 2012 patch level should be SP1 CU2.
4. SLA for processing the request.

# Execution Steps

Steps to be followed for executing the RFC for this KCT.

### Dummy names & configuration settings used in this setup.

Following server hosts are running inside a pre-existing domain & already part of a windows failover cluster.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **IP** | **Type** | **Comments** |
| W61DC | 192.168.1.1 | Domain Controller | Domain Controller for domain **Tree.com**. |
| W61NA | 192.168.1.101 | Cluster Node | First node |
| W61NB | 192.168.1.102 | Cluster Node | Second node |
| HADRclus | 192.168.1.31 | Windows Failover Cluster | This will hold the listener inside a resource group which can be failed over between nodes Wr2NA & Wr2NB. |

Domain based service accounts being used.

|  |  |
| --- | --- |
| **Account Name** | **Purpose** |
| Sqleng | Domain level service account to be used with SQL Server DB engine. |
| Sqlage | Domain level service account to be used with SQL Server Agent. |

Names we will use while configuring AO availability groups & listener.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **IP** | **Type** | **Coments** |
| HADRaoag | na | Availability Group Name | The Availability Group name which will also appear as the cluster failover resource group name inside the cluster administrator. |
| HADRlist | 192.168.1.32 | Availability Group Listener Name / CAP | This will be the name/ip through which clients will connect to the databases under Availability Group. |
| AOseed | na | User Database | This DB will be used to demonstrate AO setup. |

### Preparation Steps

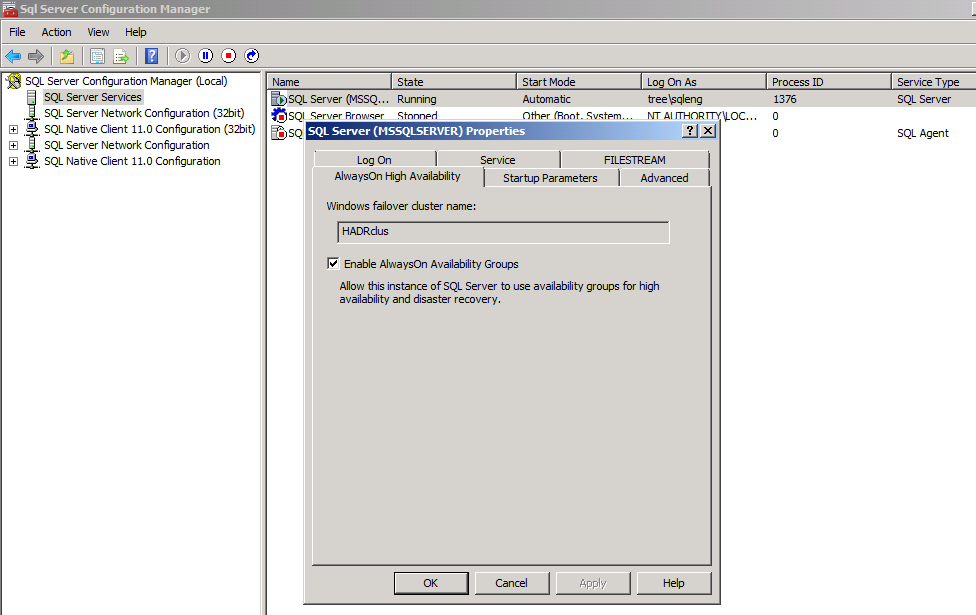
1. Identify a shared location & give RW access to SQL Engine Service Account. This is generally the backup drive on the Primary server, i.e. E: drive. Initial AOseed database is a copy of model DB otherwise for big databases there should be enough free space. In absence of which we can use the setup option join only.
2. Create firewall exceptions on all the nodes participating in the AO group. These exceptions will be inbound & named as [**sqlAG**].

TCP Ports to be identified

* + 1. **1433** 🡪 or any other port the SQL instance is designed for.
    2. **5022** 🡪 or any other port in case of multiple instances.

# In case of multiple instances on the same machine additional ports can be added to this exception list.

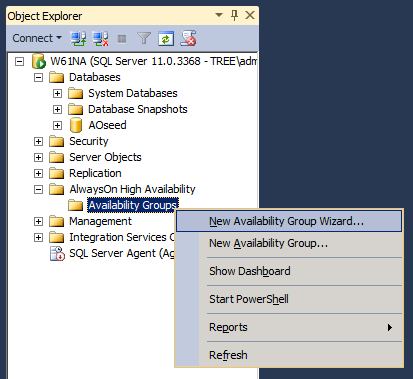
1. Enable AlwaysOn for each of the identified SQL instances using SQL Server Configuration Manager on each node. Right click on SQL Server Engine service and select Properties. Then select the AlwaysOn High Availability tab.



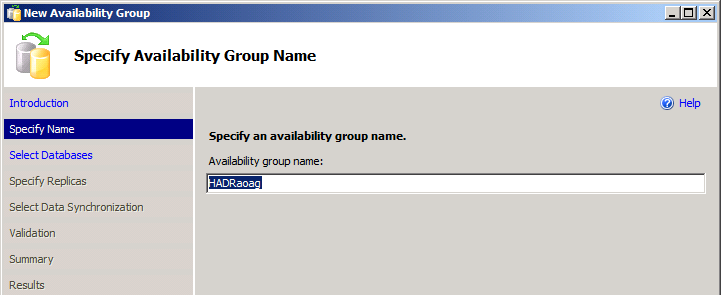
1. If using GUI the SQL Services need to be restarted manually.
2. When doing on a FCI, this needs to be done on active node only.
3. If the underlying WSFC is destroyed and recreated, then this feature will need to be Unchecked and the service restarted and then checked again and service restarted.
4. Connect to the server that will be the Primary replica using SSMS and create a seed database named AOseed. This should be in the default data and log directories, which should be the same across all instances participating in the Availability Group. The database should be owned by SA and must be in Full Recovery Model.
5. Take a full backup of the AOseed database to the default backup location.

### Availability Group Creation

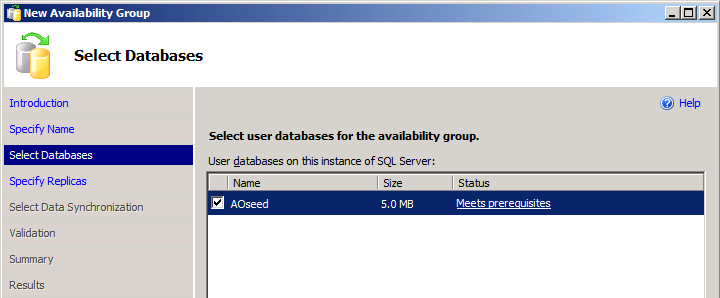
* 1. Using **New Availability Group Wizard** to create an AOAG.
  + Using SSMS connect to the server you want to make primary.
  + Right click on the Availability Groups & select New Availability Group Wizard



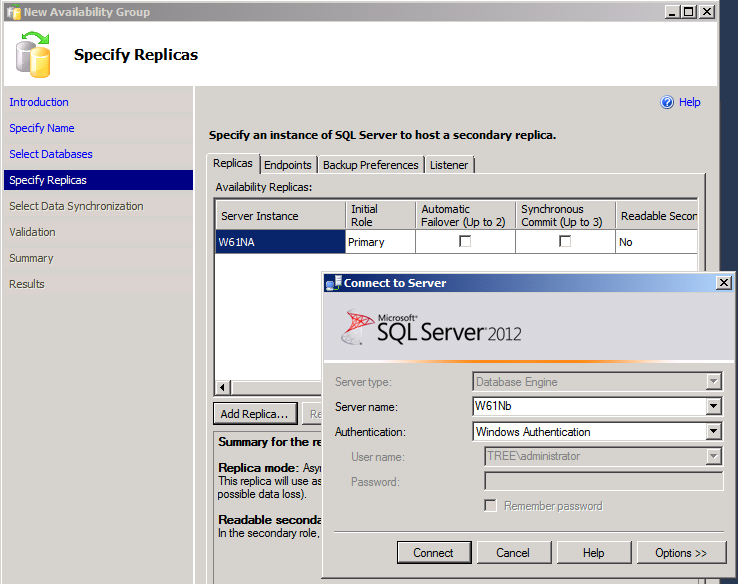
* 1. Provide an **Availability Group Name**
  + This will be the Cluster Resource group name (this is unique on the instance & cluster). It is not registered in DNS or AD.

****

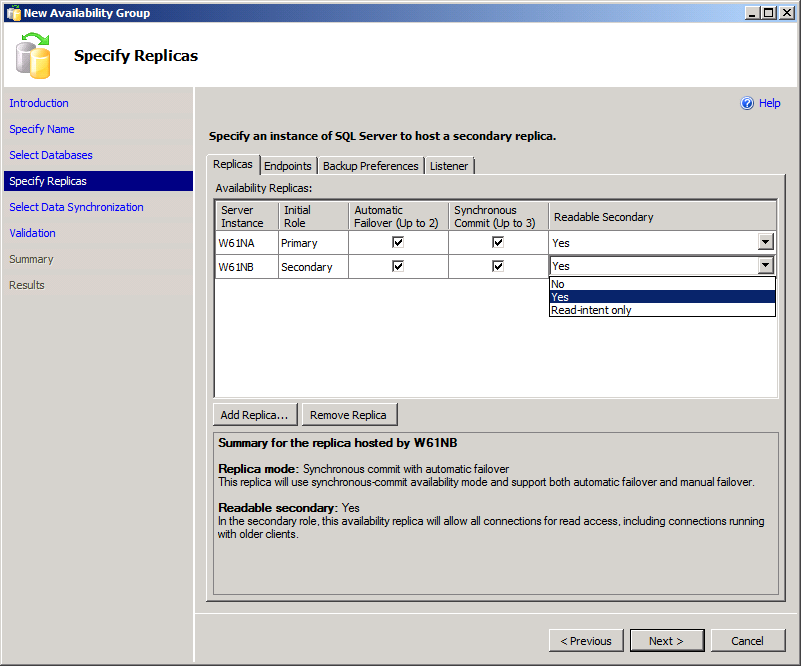
* 1. Select **Participating Databases**
  + After clicking next, select User Database(s) for Availability Group. This should be the AOseed database created previously.
  + In case the database is not listed or doesn’t says meets pre-requisites check section **Preparation Steps,** make changes & refresh.

****

* 1. Add **Replica servers**
  + Click on the add replica button to show **connect to server dialogue**. Provide the server name(s) & click connect.

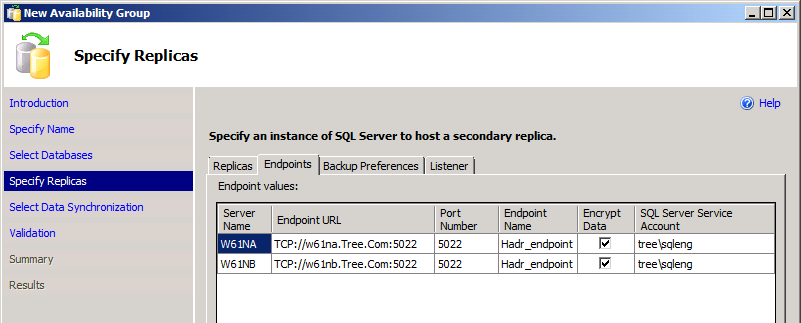


* + Once connected the screen looks something like below, here one can choose the replica mode & connection type for when the instances will be in secondary role.



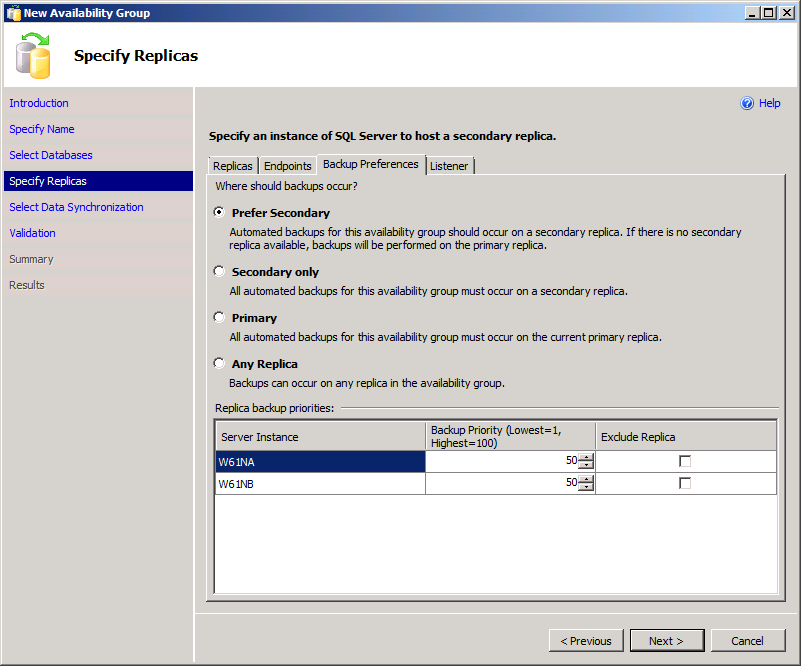
* + Some key points to note here are;
    - Host on which you are configuring **AlwaysOn** will automatically be taken as **Primary**, replicas as added will be marked secondary.
    - Total of 5 sql instances can participate in an AlwaysOn Availability Group though **only two will support Automatic Failover** & to provide for Automatic Failover the replicas should be doing Synchronous commit. At a time a **total of three out of five servers can be set to have Synchronous Commit**.
    - RO activities can be directed to the secondary replicas which will release RO load on the primary though causing a bit of overhead on secondary, the available options are
    - Yes - Allows all connections but only RO activities will be honored.
    - Read-intent only - Only connections specified as RO will be allowed.
    - Like RO activities backups (FULL with copy only & TLOG), DBCC checks can also be directed to the secondary replicas.
  1. **Configure Endpoints**

This tab shows the default endpoints selected & you‘ve the option to mark the communication as **encrypted** or otherwise. It also shows the SQL Services account names.



* + The Endpoints will be automatically created & shared amongst multiple AOAG on the same instance.
  + When setting up AOAG for multiple instances on the same machine, you will need to specify different port#.
  + For the ports listed in the endpoint url against the server name, make sure those ports are inbound open in firewall on all nodes.
  1. **Backup** **Specifications**

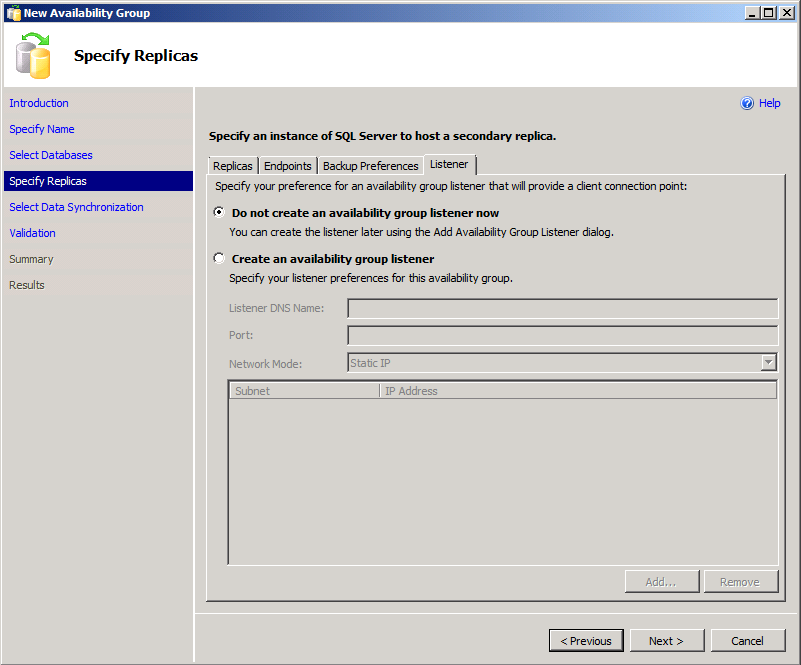
It’s suggested to have a **centralized backup** location. Backups operations can be directed to any of the participating servers or the selected ones can be excluded as well. Selection order is priority driven where 100 is highest & 1 being lowest.



After completing the setup, to determine whether the current replica is the preferred backup replica use **sys.fn\_hadr\_backup\_is\_preferred\_replica** function it just checks if (top-priority replica is local).

* 1. **Listener Configuration**

This last tab is used to configure a Listener for this availability group (SSMS can add only one listener per AOAG. In case multiple listeners are required use CLUadmin or PSH).

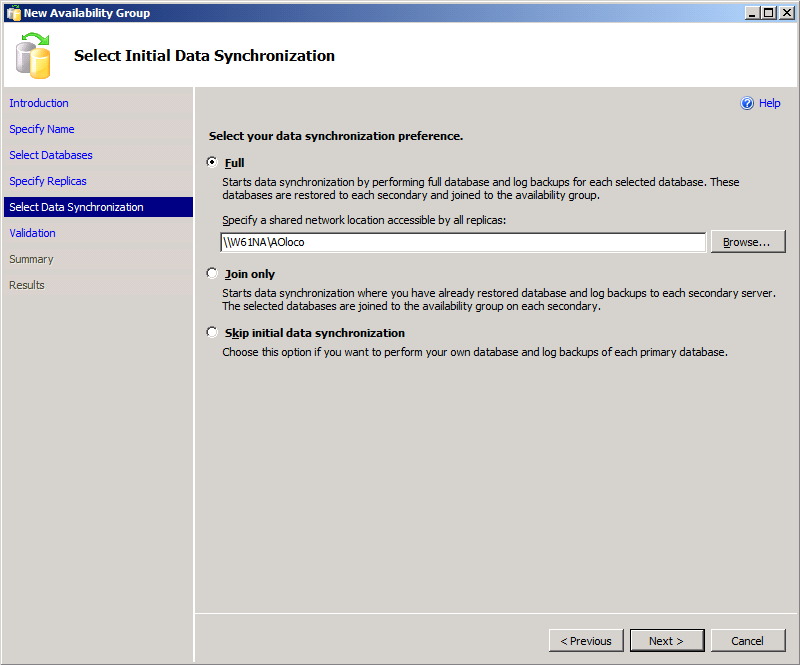


Creating a listener involves creation of new resources in cluster resource group & account creation in the Active Directory, there can be complications under AD because of duplicate names, out of range IP addresses & permission restrictions. To make it easy we will create an AOAG first & add the listener later.

* 1. **Synchronization** **Preferences**

Hitting next brings us to this screen, where we can choose how the initial data synchronization will take place.

* + Initial Data Synchronization can be done through the wizard. This requires a shared folder on primary with RW access to the SQL Service accounts. The wizard will do a full-copy only + log backup on this location & restore them on the replica, the database(s) will show as restoring unless they are synchronized.
  + This methoud is not suggested when there are multiple / big databases to be added. In such scenarios first backup & restore a FULL+LOG backup on the secondary with NORECOVERY option then select the option Join only.



**\*Avoid giving RW access to everyone, if feasible drop this folder after setup completes.**

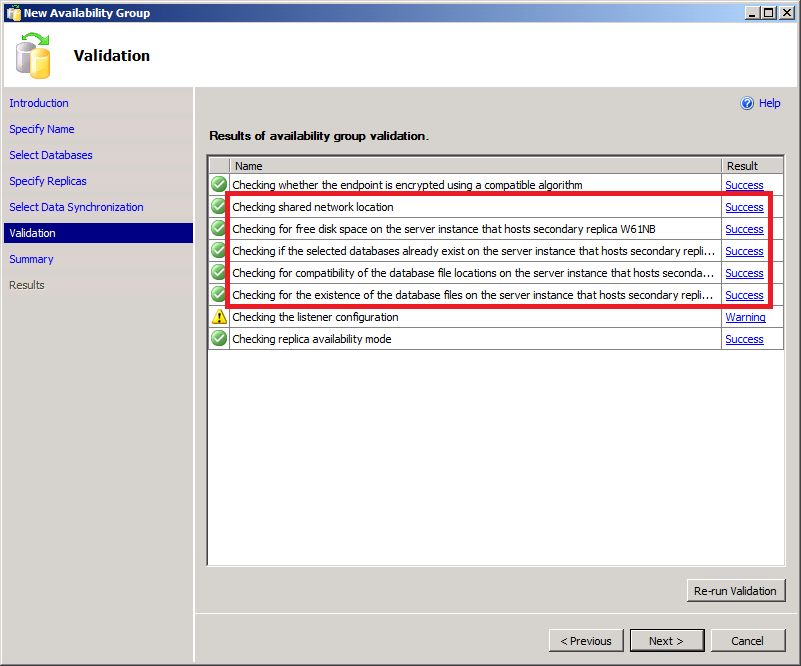
**\*Incase the shared location will stay for long & expected to be regularly used, it would be better suited on a NAS or UNC rather than one of the replicas.**

* 1. **Validations**

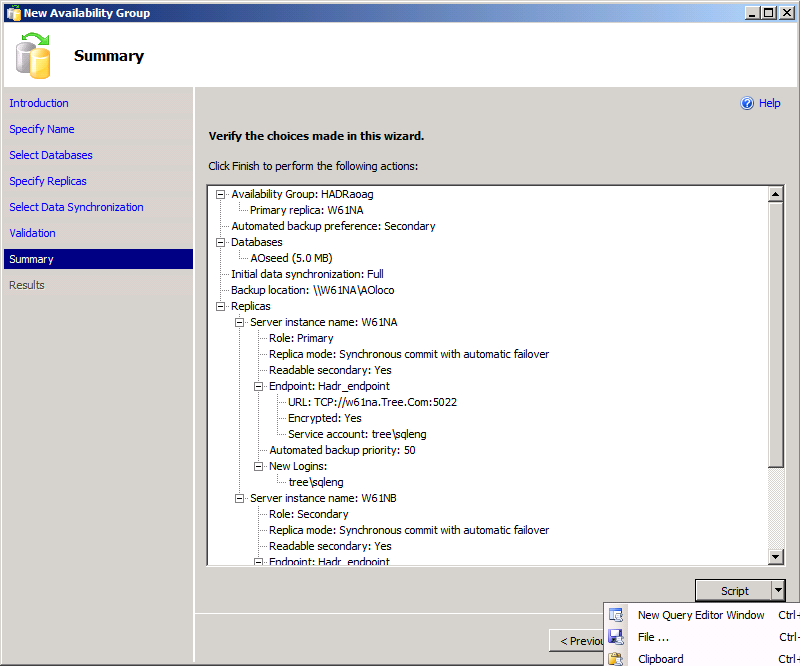
Following are checked on this validation screen.

* + Encryption of endpoints at all the instances
  + Only in case the wizard will sync the databases following are applicable
    - Verification of RW access to the shared network location
    - Checking of free disk space, database existance, symmetric storage, file name availability
  + In this case we are not creating a listener so its highiited in yellow.
  + Availibility mode of the replica

In case you select the option JOIN ONLY in the previous step, the points marked inside the red box will be skipped. The four checks after checking shared network location will be repeated for every replica added.

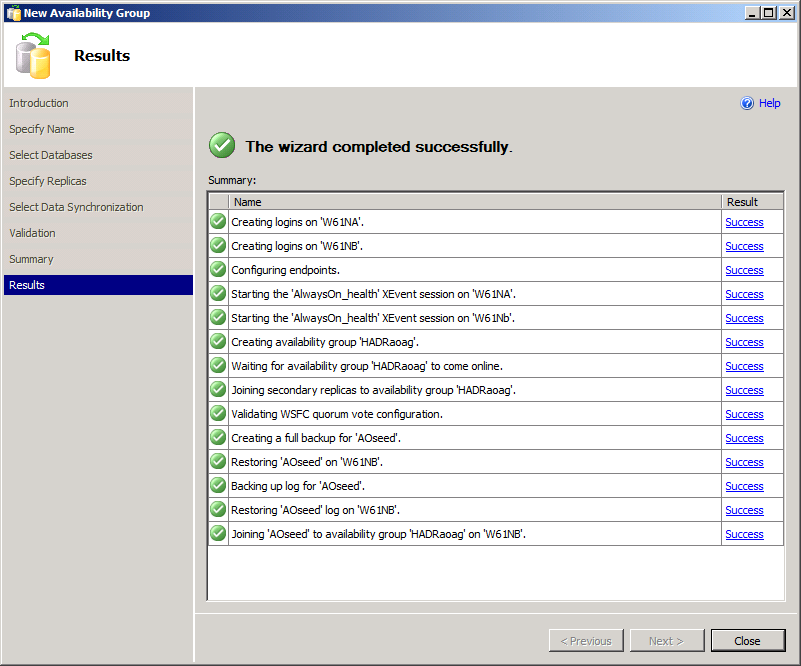


* 1. **Summary Screen**

The screen below shows a review of the options selected providing some idea about the expected actions that will be performed. These can be scripted for reference & help, if this fails. Once you’ve checked/scripted these, click next.

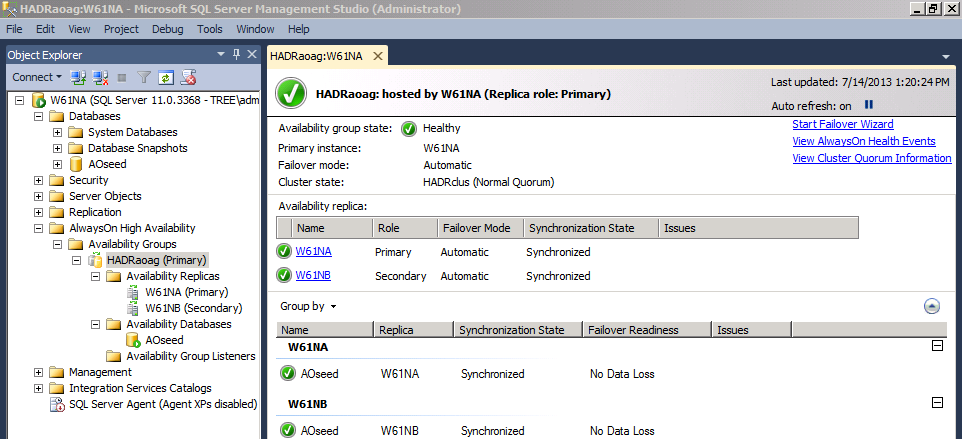
* 1. **Progress through the setup/creation of AOAG**

Progress on creation of AOAG is shown on this screen (click on the hyperlinks to the right for details).



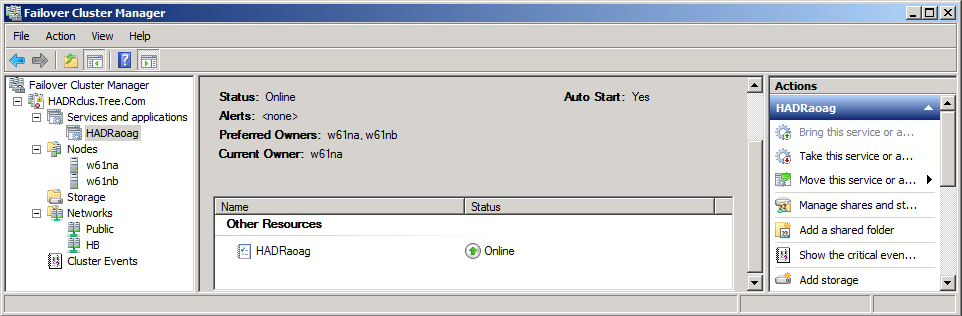
* 1. **AOAG listing in SSMS – Object Explorer**

After this point the object explorer will show the AOAG status as under, one can right click the AOAG name & open dashboard to see the synchronization status, do so on the primary replica will show the status for all the participating instances where as the same on secondary replica will show status for that that instance only.



* 1. **AlwaysOn Resources as shown under CluAdmin**

The status of the AO resource group can also be checked from Cluster Admin.



The above figure has the AOAG resource only, now we will add a listener using SSMS which will add a CAP & a dependent IP address.

* 1. **Change AlwaysOn Availability Group session timeout.**

Default value is set to 10 but in case of multi subnet scenarios, high numbers of nodes in the cluster this should be relaxed. This is to avoid the possibility of a heavily loaded system missing PINGs and declaring a false failure

|  |  |  |
| --- | --- | --- |
| Availability Group Replica Property  |  |  | | --- | --- | | Session Timeout | 20 Seconds | |

**Run once for each replica.**

ALTER AVAILABILITY GROUP <**AGName**> MODIFY REPLICA ON <**ReplicaName**> WITH (SESSION\_TIMEOUT=20)

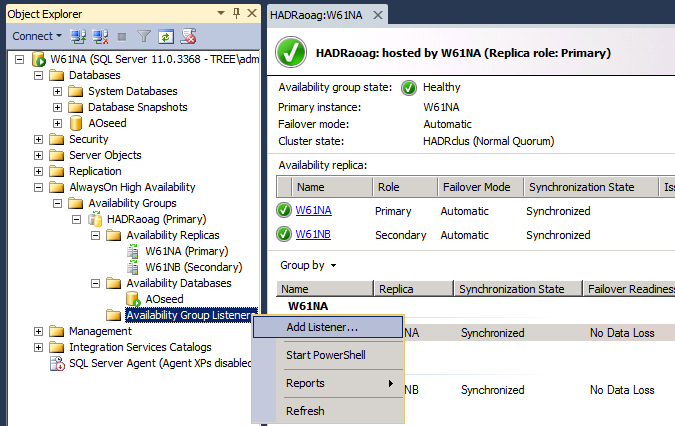
Use catalog view **sys.availability\_replicas**.**session\_timeout** to see the current set value.

### Adding a Listener to an AOAG

To create a listener we will need a unique name in AD, an available IP address and a port for communication (you may use the current SQL port#).

1. Select the **Availably Group** to **add a listener**

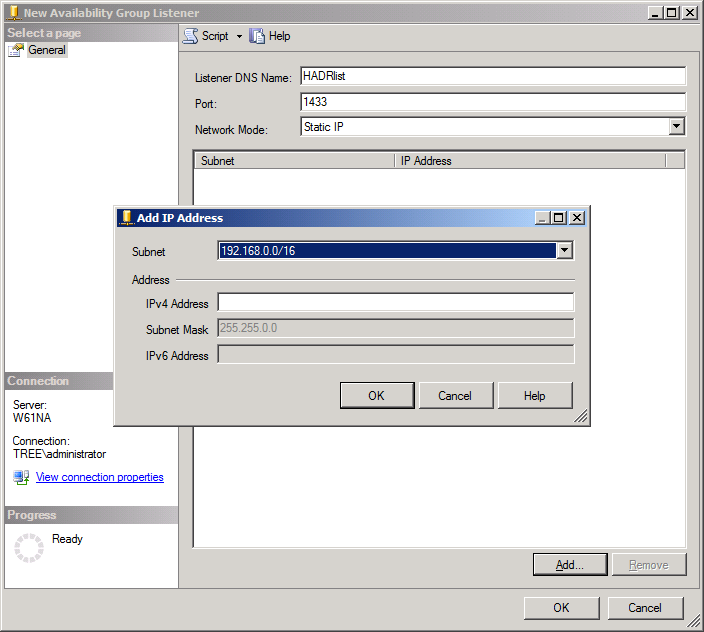
Expand the specific AOAG & right click on Availability Group Listeners, select Add Listener.



1. Provide values for **Availably Group Listener Name, IP Address & Port#**.

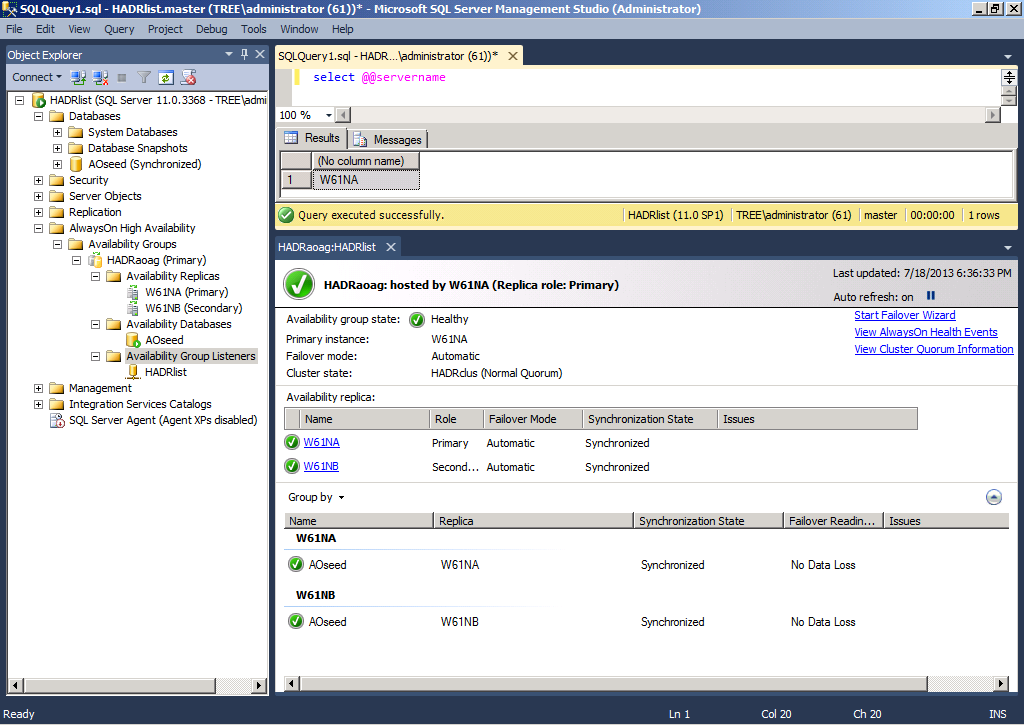
Listener configuration requires an IP address & a port number to listen to (Listener name & IP are uniquely registered in AD), though DHCP can be used but in production cases it’s suggested to have a Static IP. If port numbers other than 1433 are used they have to be mentioned in the connection string.

* + Avoid using names that are longer than 15 chars.
  + With SQL 2012 clusters & AOAG can go multi-subnet, it’s required to specify the IP address from the correct range (subnet).
  + Listener ports are not detected by the Browser Service
  + Unless required an AOAG will only host one listener & that can use the same port as being used by SQL Server.



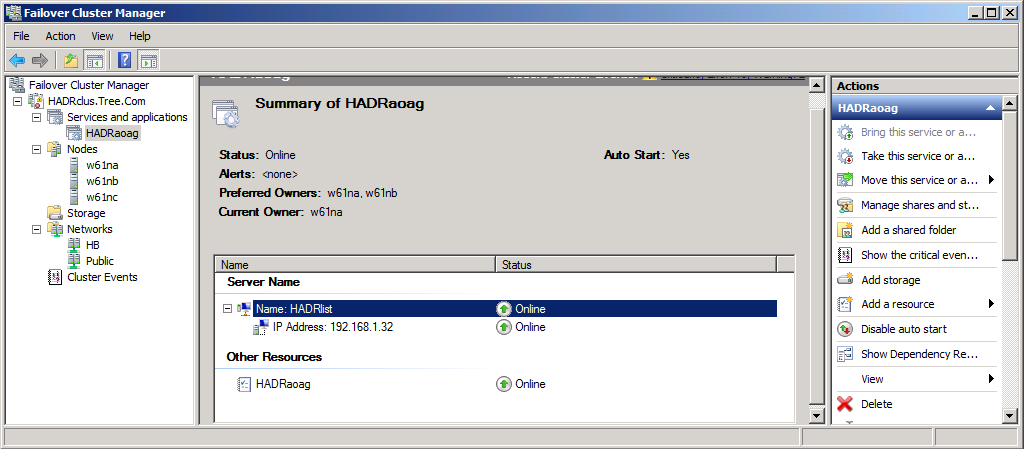
Except for the port #, no other settings for the listener can be changed later from SSMS. To rename the listener or change the IP address this has to be dropped & re-created otherwise use PowerShell or ClusterAdmin.

On clicking OK the listener will be created & hosted under the AOAG like this.



\*Any operations that affect the Active Directory objects should be planned keeping in mind the time it takes for replication, stale entries need to be manually cleared from AD.

This is how it’s shown under the cluster admin, the resources can be seen online after creation.



1. **Configure Listener Parameters**

Depending on the type of AOAG configuration selected, there are a few parameters that need to be modified in case the environment involves a multi-subnet setup (An option to avoid this to add patch # 2654347 which will provide support to SQL Client for AlwaysOn features of Read-intent, read-only & multi-subnet failover. The hotfix needs to be installed on each Reporting Services report server & the clients processing).

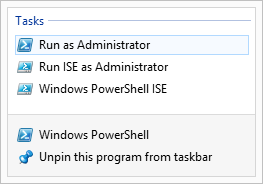
When modifying any of such parameters, it’s required to bounce (offline/online) the cluster resource for changes to take effect.

**\*Win 2012 issues a warning to bounce the resources but there are no such reminders in older versions.**

As these are Cluster Name Private Properties, they are not exposed through GUI. In the following steps PowerShell will be used to alter two such properties.

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Default Value** | **New Value** |
| HostRecordTTL | 1200 | 300 |
| Time in seconds before the IP / Name association information will be re-circulated over the network. | | |
| RegisterAllProvidersIP | 1 | 0 |
| When 0 it will not register multiple IP addresses (multi-subnet setup), only the current live IP will be associated. If set to 1 all the IP addresses associated with the Listener name will be registered, this may cause problems if the keyword **multi-subnet failover = true** is not used. For multi-subnet clustering, see [SQL Server Multi-Subnet Clustering (SQL Server)](http://msdn.microsoft.com/en-us/library/ff878716.aspx). | | |
| **Table 1** | | |

To make changes to these settings one should use PSH with administrator privileges.



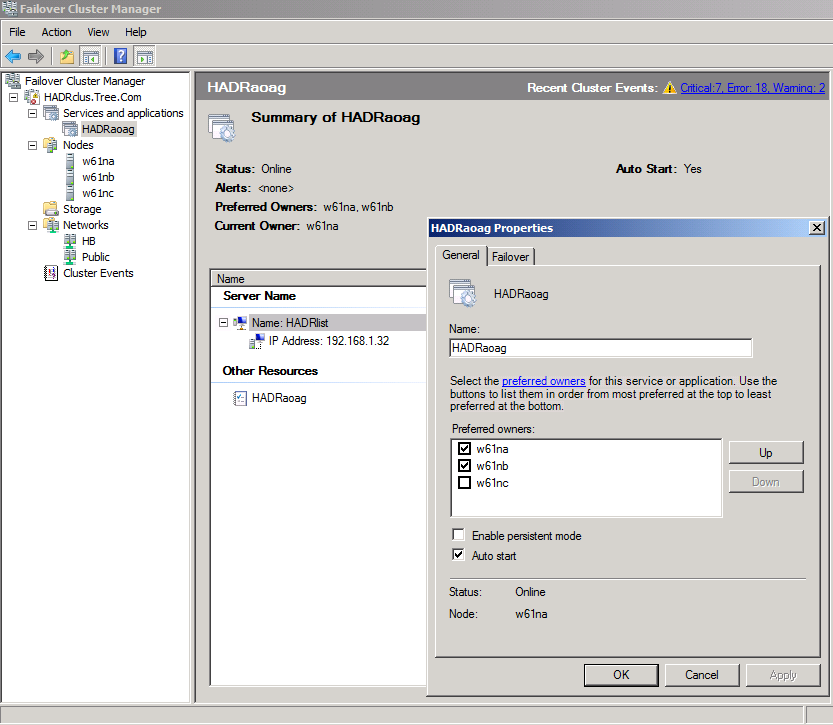
On some of the systems one may have to explicitly call in support for failover cluster commands by running **Import**-**Module FailoverClusters**.

The following commands illustrate changing the parameters listed in **Table 1**. When running these commands from PSH make sure to verify the resource names as these may not be the names seen in DNS, SSMS or CluAdmin. When created from SSMS the name will be “**AGName\_ListenerName**” & when created from CluAdmin it will be “**ListenerName**”

|  |
| --- |
| Check / verify the cluster name for the current machine. |
|  |
| List all resource groups of the cluster |
|  |
| Filter & list all resources in our AOAG. |
|  |
| Show properties of the AOAG Listener or **C**lient **A**ccess **P**oint. |
|  |
| Alter the two specific parameters. |
|  | |
| Verify the properties again. |
|  |

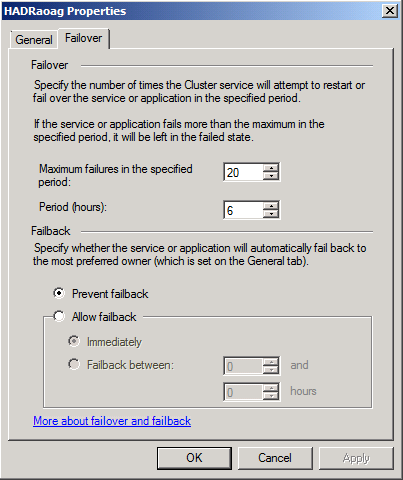
1. **Configure AOAG resource group properties for failover & fall back.**

In the figure below you can see the participating nodes of AOAG are checked (never modify OWNER settings from CluAdmin, typically in the cases when there’s a FCI instance involved. The resource will change owners as FCI bounces between nodes).



On the second tab

* + Increase the count of **Maximum failures to 20** by default this number is node count -1.
  + Select the option to **prevent failback**.



Change in these don’t require you to bounce the resource.

**At this point the AG has been built and is ready for QC and handoff.**

# Escalation Matrix

Following Escalation Metrics should follow:

1. Check in PRIMUS with error code.
2. Any issue or challenge reported need to get escalated to service owners.

# Appendix

# Document Administration Page

**Document History**

|  |  |  |
| --- | --- | --- |
| Date | Updated by | Change |
| Mar 20, 2013 | Steve Harrell | Draft of Document |
| Mar 27, 2013 | Yusuf Anis | Update & Internal Review |
| July 23, 2013 | Yusuf Anis | Modification & Update |
| July 3, 2014 | Yusuf Anis | Added Session\_TimeOut recommendations. |
|  |  |  |