# 

**AlwaysOn Architecture Guide: Building a High Availability and Disaster Recovery Solution by Using Failover Cluster Instances and Availability Groups**

SQL Server Technical Article

**Writers:** Joseph Sack (SQLskills.com), Sanjay Mishra (Microsoft)

**Technical Reviewers:** Min He (Microsoft), Chuck Heinzelman (Microsoft), Alexi Khalyako (Microsoft), Charles Mathews (Microsoft), Prem Mehra (Microsoft) , Juergen Thomas (Microsoft), Mike Weiner (Microsoft), Amitabh Tamhane (Microsoft), Brent Ozar (Brent Ozar PLF), Gianluca Hotz (SolidQ), David P. Smith (ServiceU), Michael Steineke (Edgenet), Glenn Berry (SQLskills.com)

**Content Program Manager:** Glenn Minch (Microsoft)

**Published:** June 2012

**Applies to:** SQL Server 2012

**Summary:** SQL Server 2012 AlwaysOn Failover Cluster Instances (FCI) and AlwaysOn Availability Groups provide a comprehensive high availability and disaster recovery solution. Prior to SQL Server 2012, many customers used FCIs to provide local high availability within a data center and database mirroring for disaster recovery to a remote data center. With SQL Server 2012, this design pattern can be replaced with an architecture that uses FCIs for high availability and availability groups for disaster recovery business requirements. Availability groups leverage Windows Server Failover Clustering (WSFC) functionality and enable multiple features not available in database mirroring. This paper details the key topology requirements of this specific design pattern, including asymmetric storage considerations, quorum model selection, quorum votes, steps required to build the environment, and a workflow illustrating how to handle a disaster recovery event in the new topology across participating job roles.

Copyright

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2012 Microsoft. All rights reserved.

Contents

[Introduction 4](#_Toc326897512)

[Failover Cluster Instances for local HA and Database Mirroring for DR 4](#_Toc326897513)

[Failover Cluster Instances for local HA and Availability Groups for DR 5](#_Toc326897514)

[Planning and Considerations 7](#_Toc326897515)

[Windows Server Failover Cluster Requirements 7](#_Toc326897516)

[Asymmetric Storage 7](#_Toc326897517)

[Instance Naming and File Path 8](#_Toc326897518)

[Availability Mode and Failover Mode 8](#_Toc326897519)

[Quorum Voting 8](#_Toc326897520)

[Quorum Configuration 9](#_Toc326897521)

[Tools to View/Change Quorum Model and Node Votes 11](#_Toc326897522)

[Configuring the WSFC Quorum Model 11](#_Toc326897523)

[Using DMVs and AlwaysOn Dashboard to view Quorum Information 11](#_Toc326897524)

[Configuring Node Votes 13](#_Toc326897525)

[Client Connectivity 13](#_Toc326897526)

[Read-Write Workloads 14](#_Toc326897527)

[Read-Only Workloads 14](#_Toc326897528)

[Configuring the FCI+AG Solution 15](#_Toc326897529)

[Installing Prerequisites 15](#_Toc326897530)

[Setting up the Solution at the Primary Data Center 15](#_Toc326897531)

[Setting up the Solution at the DR Data Center 20](#_Toc326897532)

[Monitoring Considerations 24](#_Toc326897533)

[Recovering from a Disaster 25](#_Toc326897534)

[Reverting Back to the Primary Data Center 31](#_Toc326897535)

[Conclusion 35](#_Toc326897536)

[References 36](#_Toc326897537)

# Introduction

Microsoft SQL Server 2012 AlwaysOn provides flexible design choices for selecting an appropriate high availability (HA) and disaster recovery (DR) solution for your application. For more information about SQL Server 2012 AlwaysOn high availability and disaster recovery design patterns, see [SQL Server 2012 AlwaysOn High Availability and Disaster Recovery Design Patterns](http://go.microsoft.com/fwlink/?LinkId=255048).

This white paper describes the solution using failover cluster instances (FCI) for HA and using availability groups (AG) for DR. This architecture combines a **shared storage** solution (FCI) and a **non-shared storage** solution (AG).

Prior to SQL Server 2012, a common HA and DR deployment architecture involved the use of FCIs for local high availability and database mirroring (DBM) for remote disaster recovery. With SQL Server 2012, availability groups can replace the database mirroring component of the solution.

This paper covers planning considerations and walks through the steps required to build this solution. It also covers the steps required to recover from a disaster, and it explains how to revert back to the primary data center after the primary data center is restored.

This paper assumes a basic knowledge of failover cluster instances (FCIs), availability groups, high availability, and disaster recovery concepts. For more information about the full AlwaysOn solution feature set, see the [Microsoft SQL Server AlwaysOn Solutions Guide for High Availability and Disaster Recovery](http://msdn.microsoft.com/library/hh781257.aspx) white paper. For more information about migration steps, see the [Migration Guide: Migrating to SQL Server 2012 Failover Clustering and Availability Groups from Prior Clustering and Mirroring Deployments](http://msdn.microsoft.com/library/hh923056.aspx) white paper.

The target audience for this white paper includes operational SQL Server database administrators and technology architects. It is also appropriate for system administrators who collaborate with the database administrator role for management of Windows Server, Active Directory Domain Services (AD DS), WSFC, and networking.

# FCIs for Local HA and Database Mirroring for DR

As mentioned in the introduction, before SQL Server 2012, a popular SQL Server deployment architecture involved the use of FCIs for local high availability, and the use of database mirroring for cross-data center disaster recovery. This was referred to as an *FCI+DBM* solution. For this solution, one FCI is configured within the primary data center using shared disk storage (via SAN, for example) in order to provide SQL Server instance-level protection. If a hardware failure occurs on one of the nodes, another node can take over as the host of the FCI within the same data center.

Database mirroring is used between the primary site and the disaster recovery site to provide database-level protection. In the event of a primary data center outage, or if the shared storage in the primary data center experiences a failure, the mirror in the DR data center can be used to restore service to the applications. The disaster recovery data center hosts another FCI on a separate WSFC, with its own shared storage. Figure 1 provides a representation of this solution architecture.

Figure 1: FCI for high availability and database mirroring for disaster recovery

Typically, the DR data center is located at a distance from the primary data center, and the mirroring session is set to “high performance” asynchronous mode in order to minimize the overhead to transactions. Occasionally, synchronous database mirroring between the data centers is also observed.

For more information, including a practical example of this specific solution, see [High Availability and Disaster Recovery at ServiceU: A SQL Server 2008 Technical Case Study](http://go.microsoft.com/fwlink/?LinkId=255052).

# FCIs for local HA and Availability Groups for DR

With SQL Server 2012, a similar solution involves using FCIs for local high availability, like the FCI+DBM solution, but using availability groups (AG) for disaster recovery. This is referred to as an *FCI+AG solution*.

Figure 2 shows the solution using FCIs for local high availability and availability groups for cross-data center disaster recovery.

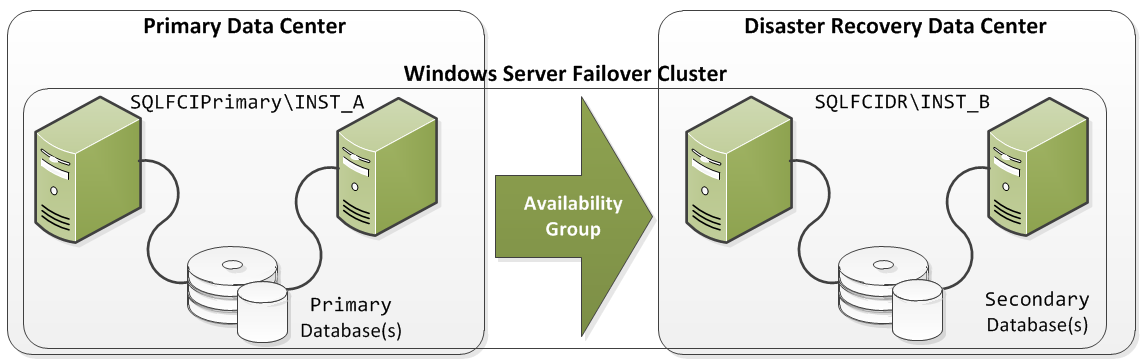


Figure 2: FCIs for high availability and availability groups for disaster recovery

Figure 2 shows two FCIs, one in the primary data center and another in the disaster recovery data center. Each FCI has two nodes and its own shared storage. All four nodes, however, are part of the *same* WSFC. That all nodes belong to the same WSFC is a requirement for availability groups.

Figure 2 illustrates a simple scenario topology with two data centers, each hosting one replica of the AG on a two-node FCI. The architecture allows for variations to this topology:

* Multiple data centers
* Multiple replicas, up to five including one primary replica and one to four secondaries
* More than two nodes in each FCI if additional passive nodes are desired for HA purposes
* Not all of the replicas in an availability group need to reside on FCI instances; some replicas can reside on non-FCI stand-alone SQL Server instances
* Multiple availability groups based on the logical grouping of databases for your application environment

The discussion in this white paper focuses on the topology shown in Figure 2; however, the general concepts apply to the other variations as well.

Because the four nodes across two sites are part of the same WSFC, there are additional considerations for using shared storage that is visible only to the local data center nodes. There are also additional considerations around quorum voting and the quorum model. This paper discusses these and other considerations.

The availability group can be configured with one or more user databases, and it can use either synchronous or asynchronous data movement. Synchronous replicas add latency to the database transactions because the primary needs to receive the acknowledgement that log records have been hardened to the secondary replica logs before the primary replica commits the transaction.

It is also important to note that the disaster recovery SQL Server instance does not need to be an FCI. An availability group could also have a stand-alone SQL Server instance for the secondary replica. With availability groups, you can mix both stand-alone instances and FCIs within a single topology on the same WSFC. Figure 3 shows a mixed topology.



Figure 3: FCI for local HA and availability groups for DR, with the DR instance being a stand-alone instance

The rest of the paper assumes that both the primary and secondary replicas are hosted FCIs, and not stand-alone instances.

# Planning and Considerations

This section details planning considerations, requirements, and prerequisites to consider before you implement an FCI+AG solution for high availability and disaster recovery.

## Windows Server Failover Cluster Requirements

A fundamental change between an FCI+DBM versus an FCI+AG solution is that you are moving from using two FCIs on two separate WSFCs to using two FCIs on a *single* WSFC. All replicas for an availability group must exist on a *single* WSFC within a single Active Directory domain, even between data centers.

## Asymmetric Storage

Two FCIs, one at each site on a single multi-site WSFC, introduce considerations around how shared storage is handled. Each FCI has its own shared storage. The nodes at the primary site share storage among themselves to form a shared-storage FCI, and the nodes at the DR site share storage among themselves to form another shared-storage FCI. The storage on the primary site is not visible to the nodes on the disaster recovery site and vice versa. This arrangement of storage, where a cluster disk is shared between a subset of nodes within a WSFC, is referred to as *asymmetric storage*. Before the asymmetric storage capability, shared storage needed to be visible to *all* the nodes in the WSFC (symmetric storage). Asymmetric storage was introduced as a deployment option for Windows Server 2008 via a hotfix. Asymmetric storage is also supported in Windows Server 2008 R2 via Service Pack 1. For more information about this hotfix, see the Knowledge Base article [Hotfix to add support for asymmetric storages to the Failover Cluster Management MMC snap-in for a failover cluster that is running Windows Server 2008 or Windows Server 2008 R2](http://support.microsoft.com/kb/976097).

This Windows Server enhancement is the key piece of functionality that enables the FCI + AG solution architecture discussed in this white paper. By enabling this functionality, you can combine the shared storage solution (FCI) with the non-shared storage solution (availability groups), in a single HA + DR solution. Consequently, this enhancement also enables you to use identical drive letters for shared disk resources across data centers.

Note that when you configure asymmetric storage, you may see a message during WSFC validation tests that “*Disk with id XYZ is visible or cluster-able only from a subset of nodes*”. For asymmetric storage, this is expected and not a cause for concern.

## Instance Naming and File Path

The two FCIs must use different instance names within the same WSFC, for example, using “INST\_A” as the instance name for the primary FCI and “INST\_B” as the instance name for the DR FCI. (In contrast to availability groups, database mirroring permits each FCI to use the same instance name if the FCIs are on separate WSFCs. In Figure 1, both FCIs used the same the same instance name, INST\_A, with the FCI+DBM solution).

Each FCI has its own shared storage, which is not accessible by nodes in the other data center, and which should use identical drive letters for the disks, as well as identical file paths for database files and transaction log files in both of the FCIs. Identical file path and drive letters are not an absolute requirement, but if file paths differ, you will need to do a manual RESTORE WITH MOVE when you restore the replica databases on the secondary. Moreover, heterogeneous paths across the two FCIs will invalidate later file addition operations, such as creating file groups or secondary log or data files. For more information, including a problem scenario and resolution, see [Troubleshoot a Failed Add-File Operation (AlwaysOn Availability Groups)](http://msdn.microsoft.com/library/hh510190).

## Availability Mode and Failover Mode

For the availability group created between the two FCIs, you can designate either synchronous or asynchronous commit availability modes. If the availability mode is synchronous, the primary replica waits to commit user transactions until they have been sent to and hardened on the secondary replicas. This can add latency to the user transactions but also helps you to eliminate the chance of data loss to the secondary replica by ensuring that transactions are sent to the disaster recovery FCI before a commit is signaled on the primary replica transaction.

If the availability mode is asynchronous, your primary replica user transactions do not wait for transactions to harden on the secondary replica logs. This reduces transaction latency, but it increases the exposure to data loss in the event of an outage.

Regarding failover modes, when FCIs are used in the availability group topology, the availability groups’ failover mode must be manual (not automatic). However, within each FCI, the FCI failover of the SQL Server instance to other nodes is automatic.

## Quorum Model and Node Votes

**Note**: Discussions of quorum models and related information in this white paper apply to solutions running on Windows Server 2008 and Windows Server 2008 R2 operating systems, with appropriate service packs and other software updates.

Because the underlying infrastructure of the FCI+AG solution is a WSFC, it is important to consider appropriate quorum model for the WSFC. Quorum configuration is managed at the WSFC level, irrespective of the number of FCIs, the number of replicas, and the number of availability groups hosted in the WSFC.

In WSFC, there are four quorum models: **Node Majority**, **Node and File Share Majority**, **Node and Disk Majority**, **No Majority: Disk Only**. For more information about quorum models, see [Failover Cluster Step-by-Step Guide: Configuring the Quorum in a Failover Cluster](http://technet.microsoft.com/library/cc770620(v=WS.10).aspx).

Before you select a quorum model, it is important to take into consideration the number of voting nodes. Assigning appropriate node votes plays an important role in the HA+DR design. By default, every node in a failover cluster has a vote, but that may not be appropriate for your particular HA+DR solution, depending upon the distribution of nodes in the primary and DR data centers. There is a hotfix available (<http://support.microsoft.com/kb/2494036>) that allows you to assign 1 vote to some nodes and 0 votes to some other nodes in the WSFC. The NodeWeight property of the WSFC node represents the vote for that particular node. The value ‘0’ means the node doesn’t have a vote. The value ‘1’ means the node has a quorum vote. This hotfix must be installed on each node in the topology.

General recommendations for quorum voting for an AlwaysOn HA+DR solution are provided in the [Recommended Adjustments to Quorum Voting](http://msdn.microsoft.com/library/hh270280.aspx#RecommendedAdjustmentstoQuorumVoting) topic in SQL Server Books Online. These should be treated as guidelines for deciding on the voting scheme for the AlwaysOn solution. Taking these guidelines into consideration, to ensure that the quorum of the nodes in the primary data center is not compromised by outages in the DR data center or loss of connectivity between the two data centers, for the FCI+AG solution presented in Figure 2, the voting scheme will be:

* 1 vote to each node in the primary data center
* 0 votes to each node in the disaster recovery data center

This vote assignment translates to a total of 2 votes for the WSFC. As a best practice, the total number of votes for the WSFC should be an odd number. If there is an even number of voting nodes (as in our example topology), consider adding a file share witness, and then choose the Node and File Share Majority quorum model.

**Note:** In many enterprise environments it is common for a file share to be owned and managed by a different team. That team then has control over a node vote, and thus has influence on the status of the failover cluster. A file share becomes a vote and so it needs to always be available. Clustering or other HA technologies are recommended in order to ensure the availability of the file share vote.

Alternatively, you can add an additional node and use the Node Majority quorum model. The additional node needs to be within the WSFC but it does not need to be a part of the FCI configuration. It should also be located in the same primary data center, collocated with the other two WSFC nodes that exist in that data center.

Figure 4 shows the vote allocation using the Node and File Share Majority quorum model.

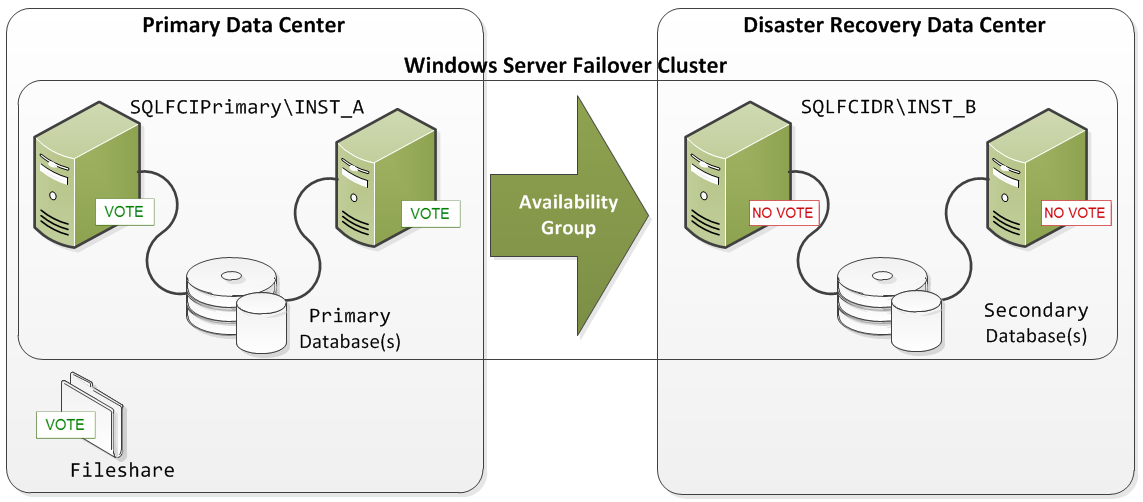


Figure 4: FCI+AG HA/DR solution with node vote assignments

In Figure 4, each of the two nodes in the primary data center has a vote. A file share witness is also defined in the primary data center and also has a vote. The two nodes in the disaster recovery data center are not given a vote and cannot affect quorum.

Additional possible quorum model choices for this deployment architecture are Node and Disk Majority (using an asymmetric disk) or No Majority: Disk Only (using an asymmetric disk). Before asymmetric storage was available in a WSFC, a shared disk could act as a quorum resource if it was visible from *all* the WSFC nodes. With asymmetric storage, cluster storage can be visible to a subset of nodes and still be used as a quorum resource. With asymmetric No Majority: Disk Only quorum model, you can implement a “last man standing” scenario, where the WSFC retains quorum as long as a single node has contact with the asymmetric disk that is acting as the quorum resource.

You can enable this using the cluster.exe command line – but you cannot enable this through Failover Cluster Manager or Windows PowerShell. For an example of this configuration, see the [Changing the quorum configuration in a failover cluster with asymmetric storage](http://technet.microsoft.com/library/cc770620(v=ws.10).aspx#BKMK_asymmetric) section of the article [Failover Cluster Step-by-Step Guide: Configuring the Quorum in a Failover Cluster](http://technet.microsoft.com/library/cc770620(v=WS.10).aspx).

**Important:** Using an asymmetric disk as the quorum resource provides numerous benefits, but it also requires a much higher level of cluster expertise and planning. You should become very familiar with this configuration before deploying it in a production environment.

In the event of a primary data center outage that requires you to bring up service in the disaster recovery data center, you must re-evaluate the quorum configuration. Each node in the disaster recovery data center must be assigned a vote and each node in the primary data center must have its vote removed (set to “0”) until service is restored. Assuming two nodes for the FCI and a longer-term outage of the primary data center, you should also configure a file share witness (or other additional vote) in the DR data center and set the quorum model accordingly. After the primary data center is ready for activity again, the voting must again be adjusted and the quorum model re-evaluated. Later in this paper we’ll step through a disaster recovery scenario and associated process flow.

The quorum model and vote assignments presented in Figure 4 assume that the solution has two replicas—one in each of the two data centers. If you have more data centers and you plan to put some part of your solution in a third data center, the quorum model decisions and vote assignments may vary.

### Tools to View and Change Quorum Model and Node Votes

There are multiple ways the cluster quorum model and/or the quorum votes can be viewed and changed. The following table lists the various tools for these tasks.

|  |  |
| --- | --- |
| **To view the quorum model** | **To change quorum model** |
| Windows Failover Cluster Manager Windows PowerShell Cluster.exe SQL Server DMVs AlwaysOn Dashboard in SQL Server Management Studio | Windows Failover Cluster Manager Windows PowerShell Cluster.exe  **Note**: Only Cluster.exe can be used to set quorum model to “Node and (asymmetric) Disk Majority” or “No Majority: (asymmetric) Disk Only” |

|  |  |
| --- | --- |
| **To view node votes** | **To change node votes** |
| Windows PowerShell Cluster.exe SQL Server DMVs AlwaysOn Dashboard | Windows PowerShell Cluster.exe |

### Configuring the WSFC Quorum Model

Here are examples of using Windows PowerShell via the command line to view the current quorum model and to change the quorum model.

**To view the existing quorum model**

Get-ClusterQuorum

**To configure the Node Majority quorum model**

Set-ClusterQuorum -NodeMajority

**To change the quorum model to Node and File Share Majority**

Set-ClusterQuorum -NodeAndFileShareMajority \\FileShare\Witness

The witness file share you choose must *not* be on a node already participating in the AlwaysOn WSFC configuration. However, it can be placed as a share on another WSFC configuration. It must exist within the same Active Directory domain as the WSFC. Also, the WSFC cluster service account requires read and write permissions to the file share witness. The Failover Cluster Manager has the built-in logic to add these permissions to the file share witness as long as the account through which the quorum model is changed has permissions on the file share.

### Using DMVs and AlwaysOn Dashboard to View Quorum Information

Though you cannot set or change the quorum model or node votes through SQL Server tools, you can use Transact-SQL queries on DMVs and use the AlwaysOn Dashboard in SQL Server Management Studio to view the node votes and quorum model of the Windows cluster hosting the availability group.

To view the quorum model of the Windows cluster hosting the availability group, query the DMV [sys.dm\_hadr\_cluster](http://technet.microsoft.com/en-us/library/hh212952(v=sql.110).aspx) (http://technet.microsoft.com/en-us/library/hh212952(v=sql.110).aspx).

SELECT cluster\_name, quorum\_type\_desc, quorum\_state\_desc

FROM sys.dm\_hadr\_cluster;

When this query is run on the example that is covered in this white paper, it returns the following.

cluster\_name quorum\_type\_desc quorum\_state\_desc

------------ ---------------- -----------------

contosocluster NODE\_AND\_FILE\_SHARE\_MAJORITY NORMAL\_QUORUM

To view the node votes, query the DMV [sys.dm\_hadr\_cluster\_members](http://technet.microsoft.com/library/hh231519(v=sql.110).aspx).

SELECT member\_name, number\_of\_quorum\_votes

FROM sys.dm\_hadr\_cluster\_members;

When this query is run on the example that is covered in this white paper, it returns the following. Vote allocation will be covered in a later section.

member\_name number\_of\_quorum\_votes

------------------ ----------------------

PrimaryNode1 1

PrimaryNode2 1

DRNode1 0

DRNode2 0

File Share Witness 1

The AlwaysOn Dashboard in SQL Server Management Studio can also be used to display quorum votes and the cluster state. Figure 5 shows this information for a Windows cluster with the Node Majority quorum model (cluster state and quorum votes are highlighted).

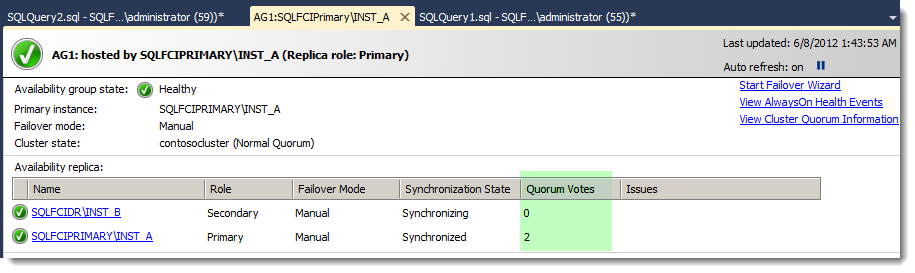


Figure 5: Displaying quorum votes and cluster state in the AlwaysOn Dashboard

Although the **Quorum Votes** column is not displayed by default, you can add it to the dashboard by right-clicking the **Availability replica** table column header and then selecting the specific column to display.

For a Node and File Share Majority quorum model, this AlwaysOn dashboard view shows only the nodes, not the file share. To see the complete quorum information, on the right, click **View Cluster Quorum Information**. A pop-up window similar to Figure 6 appears.

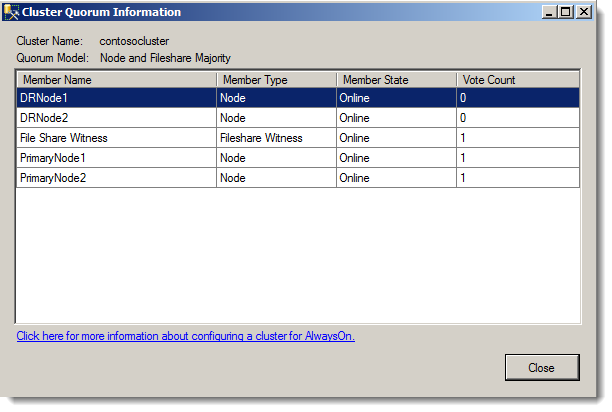


Figure 6: Cluster quorum information for Node and File Share Majority Quorum model

### Configuring Node Votes

The NodeWeight property of the WSFC node represents the vote for that particular node. The following examples demonstrate how to configure the node NodeWeight from a node in a WSFC using Windows PowerShell. For executing Windows PowerShell on the server node, click **Start**, point to **Administrative Tools**, and then click **Windows PowerShell Modules**. In this example, DRNode1 represents a specific WSFC node located in the disaster recovery data center.

**To view current vote settings for all nodes**

Get-ClusterNode | fl NodeName, NodeWeight

**To set a node’s vote to “0”**

(Get-ClusterNode "DRNode1").NodeWeight=0

**Note**: The value ‘0’ means the node doesn’t have a vote. The value ‘1’ means the node has a quorum vote.

## Client Connectivity

FCI connection methods are the same in SQL Server 2012 as they were in previous versions, but for migrations from database mirroring to availability groups, there are changes that you must consider and plan for before you can use the new readable secondary functionality. For more information about migration, including in-depth considerations and steps, see the [Migration Guide: Migrating to SQL Server 2012 Failover Clustering and Availability Groups from Prior Clustering and Mirroring Deployments](http://msdn.microsoft.com/library/hh923056.aspx) white paper.

### Read/Write Workloads

For read/write workloads that run against the availability databases in an availability group, you can connect to the primary replica using two options. The first option is to connect directly to the FCI virtual network name (VNN); each replica has a different FCI VNN. The second option is to use the availability group listener name. The availability group listener is the preferred option because it provides transparency and automatic redirection to the current primary replica, and the name in the connection string stays the same for all instances. The availability group listener is a VNN that is bound to one or more TCP/IP addresses and listener ports and is used to automatically connect to any replica without the need to explicitly designate each possible availability group replica in the connection string.

If you are migrating read/write workload application connections from a database mirroring solution that uses the Failover Partner attribute, you can still use your database mirroring connection string, but only if the availability group is configured with a *single* secondary replica that is configured for read/write activity. You can then use the initial primary replica server name as the data source and (optionally) the secondary replica name as the failover partner. This should not be used as a long-term solution, however.

### Read-Only Workloads

For read-only workload connections, you also have two options available to you. You can use the FCI VNN or you can use the availability group listener and specify the new ApplicationIntent attribute in the connection string as “ReadOnly”.

If you are using a legacy database mirroring connection string, you can only connect to the availability group as long as the availability group is configured as a single secondary replica that is configured for read/write activity.

If you want to leverage read-only routing, you must use the availability group listener name in conjunction with the ApplicationIntent attribute and “ReadOnly” value. You must also reference an availability database within the availability group. The availability group must also be configured for read-only routing to readable secondary replicas via the creation of read-only routing URLs and read-only routing lists. For more information about this process, see [Configure Read-Only Routing for an Availability Group (SQL Server)](http://msdn.microsoft.com/library/hh710054.aspx).

### Multi-Subnet Connection Support

The availability group listener can also leverage the MultiSubnetFailover connection attribute in client libraries. It is recommended that availability group connection strings designate the MultiSubnetFailover attribute for multi-subnet topologies when they reference an availability group listener name. The MultiSubnetFailover connection option enables support for multi-subnet connections and opens up TCP sockets for the availability group listener IP addresses *in parallel*. For legacy client libraries that do not support the MultiSubnetFailover attribute, you should consider appropriate client login timeout.

For more information about client connectivity and application failover considerations, see [Client Connectivity and Application Failover (AlwaysOn Availability Groups)](http://msdn.microsoft.com/library/hh213417(v=sql.110).aspx) in SQL Server Books Online.

# Configuring the FCI+AG Solution

This workflow describes the steps required to build the FCI+AG solution. Although each granular step is not described here, the goal of this section is to help clarify the workflow implementation steps and the tasks for each participating job role. Supporting documentation is referenced where appropriate. The steps are broken out by job role because most large enterprise environments define a separation of duties among the database administrator, Windows (or cluster) administrator, and network administrator roles. Because of this, it is important to properly communicate and coordinate activities across the disparate roles.

## Installing Prerequisites

Before you deploy your AlwaysOn Availability Groups solution, it is important to verify that your system meets requirements, including updates. For more information about prerequisites for deploying an AlwaysOn Availability Groups solution, see [Prerequisites, Restrictions, and Recommendations for AlwaysOn Availability Groups (SQL Server)](http://msdn.microsoft.com/library/ff878487(v=sql.110).aspx). We strongly recommend that you review this topic before you proceed.

All nodes must have the same version of the Windows Server operating system and software updates installed. The server operating system should be a minimum of Windows Server 2008 SP2, or Windows Server 2008 R2 SP1 with at least the following updates:

* Asymmetric storage (if you are using Windows Server 2008): <http://support.microsoft.com/kb/976097>
* Node votes: <http://support.microsoft.com/kb/2494036>
* Validate disk test during cluster validation: <http://support.microsoft.com/kb/2531907>

You may need additional updates.

## Setting up the Solution at the Primary Data Center

Table 1 provides the workflow for setting up the *primary data center* nodes, and it assumes that there are two nodes.

| Step | Database administrator | Windows Server \  cluster administrator | Network administrator |
| --- | --- | --- | --- |
| 1. Add the Failover Clustering feature to the two nodes located in the primary data center. For more information about this process, see [Install the Failover Clustering Feature](http://technet.microsoft.com/library/cc770506.aspx). For more information about validating your network infrastructure and other requirements, see [Understanding Requirements for Failover Clusters](http://technet.microsoft.com/library/cc771404.aspx). | Yes (coordination of activities across roles) | Yes |  |
| 1. Review the required prerequisites and install the necessary Windows Server software updates on each node in the primary data center. |  | Yes |  |
| 1. Ensure that the shared storage volumes designated for the primary data center FCI are formatted and provided with a drive letter.   We recommend that the corresponding drive letters and directory path for the DR FCI match those in the primary FCI. Keep this consideration in mind while assigning drive letters in the primary FCI. |  | Yes |  |
| 1. Validate that the account you will be using to install and configure the WSFC is a domain account. This account should also have administrator permissions on each cluster node and **Create Computer objects** and **Read All Properties** permissions for the container used for the domain computer accounts.   Alternatively, you can prestage the name object accounts ahead of time or use a domain administrator account for the installation. For more information about required permissions and provisioning options, see [Failover Cluster Step-by-Step Guide: Configuring Accounts in Active Directory](http://technet.microsoft.com/library/cc731002(WS.10).aspx). |  | Yes |  |
| 1. Using Failover Cluster Manager, perform cluster validation of the two server nodes in the primary data center and the shared storage that will be added to the WSFC. Perform the validation iteratively until there are no blocking errors.   If you are permitted to continue to the next step with the existing warnings, you need to understand all warnings to help ensure a stable configuration. For more information about performing a validation test, see [Validating a Failover Cluster Configuration](http://technet.microsoft.com/library/cc772055.aspx). |  | Yes | Yes – for any issues that may arise for the networking of the nodes |
| 1. At the end of the cluster validation step, use Failover Cluster Manager to create a two-node WSFC. For more information, including a detailed overview of this process, see [Create a New Failover Cluster](http://technet.microsoft.com/library/cc755129.aspx). |  | Yes | Yes – for any issues that may arise for the networking of the nodes |
| 1. Ensure that there are an odd number of votes; for example, you can do this by adding a file share or additional node as discussed earlier in this paper.   If you choose Node and File Share Majority, before you change the configuration, be sure that you have granted read and write permissions on the witness file share to the WSFC cluster account. |  | Yes |  |
| 1. Ensure that the installation uses shared, formatted storage that is only accessible by the two nodes located in the primary data center. These disks will be used for SQL Server in the next step. |  | Yes |  |
| 1. Install an FCI instance of SQL Server 2012 Enterprise in the primary data center. For more information, see [Create a New SQL Server Failover Cluster](http://msdn.microsoft.com/library/ms179530(v=sql.110).aspx).   You must perform two installations – the first one is **New SQL Server failover cluster installation**, which creates the FCI, and the second one is **Add node to a SQL Server failover cluster** on the second node in the primary data center. | Yes |  |  |
| 1. After you install the first FCI, enable AlwaysOn Availability Group capabilities for both SQL Server instances.   For more information about using SQL Server Configuration Manager or alternatively, SQL Server PowerShell, see [Enable and Disable AlwaysOn Availability Groups](http://msdn.microsoft.com/library/ff878259(v=sql.110).aspx). Note that when you enable AlwaysOn Availability Groups for the instance, you must restart the instance for the change to take effect. |  |  |  |
| 1. After you enable the DR FCI to support AlwaysOn Availability Groups, back up your production user databases from the legacy topology and then restore them to the primary data center FCI.   **Note**: You can choose to defer this step until the DR FCI is also available and the availability group can be set up with two replicas.  You must also script out other SQL Server objects from the legacy topology that your user databases will depend on, but that are not contained within the restored user databases (for example, SQL Server logins, associated server-level permissions, SQL Server Agent jobs).  This is similar to the process you follow when you script dependent objects that are external to the mirrored database for a database mirroring partnership. There are several methods that you can use to transfer database objects and principles between SQL Server instances. The Integration Services Transfer SQL Server Objects Task is one such method. Another method, in which logins and passwords are transferred between instances, is described here: <http://support.microsoft.com/kb/918992/> |  |  |  |

Table 1: Building the FCI+AG solution in the primary data center

## Setting Up the Solution at the DR Data Center

This table provides the workflow for setting up the *secondary*, disaster recovery data center nodes and creating the availability group.

| Step | Database administrator | Windows Server \  cluster administrator | Network administrator |
| --- | --- | --- | --- |
| 1. Add the Failover Clustering feature to all the nodes that are located in the disaster recovery data center and that participate in the solution. | Yes (coordination of activities across roles) | Yes |  |
| 1. Review the required prerequisites and install the necessary Windows Server software updates on each node in the DR data center. |  | Yes |  |
| 1. Validate that the account you will be using to install and configure WSFC is a domain account. This account should also have administrator permissions on each cluster node and **Create Computer objects** and **Read All Properties** permissions for the container used for the domain computer accounts. If you are using the same accounts as in the primary data center these permissions are already set correctly. |  | Yes |  |
| 1. Using Failover Cluster Manager, perform cluster validation of the two server nodes and shared storage joining to the existing WSFC. If you see the asymmetric storage warning message “Disk with id XYZ is visible or cluster-able only from a subset of nodes”, you do not need to take action; that this is expected and acceptable for asymmetric storage. Perform the validation iteratively until there are no blocking errors. |  | Yes | Yes–for any issues that may arise for the networking of the nodes |
| 1. After validation is finished, use Failover Cluster Manager to add the two disaster recovery nodes to the existing WSFC. |  | Yes | Yes–for any issues that may arise for the networking of the nodes |
| 1. Set the NodeWeight of the disaster recovery data center WSFC nodes to a 0 (zero) weight (see Figure 4: Cross-data center node vote assignment for an example). |  | Yes |  |
| 1. This installation should use shared, formatted storage that is only accessible by the two nodes located in the DR data center. These disks will be used for SQL Server in the next step.   Keep the drive letter and mapping identical to simplify the deployment of the availability group in later steps and allow for database file operations that do not require manual intervention or the breaking of the availability group session. |  | Yes |  |
| 1. Move available storage to one of the nodes in the DR data center for use in the next step. |  | Yes |  |
| 1. Install an FCI instance of SQL Server 2012 Enterprise in the disaster recovery data center.   You need to perform the **New SQL Server failover cluster installation** option on one of the nodes, which creates the FCI, and then perform the **Add node to a SQL Server failover cluster** option on the second node in the DR data center. | Yes |  | Yes–to coordinate the IP address (if you are using static IP addresses) and port considerations |
| 1. After installation of the two FCIs, the next step is to enable AlwaysOn Availability Group capabilities on the DR data center SQL Server instance.   For detailed steps on using SQL Server Configuration Manager or alternatively, PowerShell, see [Enable and Disable AlwaysOn Availability Groups](http://msdn.microsoft.com/library/ff878259(v=sql.110).aspx). Note that enabling AlwaysOn Availability Group for the instance will require restarting the instance to take effect. | Yes |  |  |
| 1. Script out other SQL Server objects from the legacy topology that your user databases will depend on, but that are not contained within the restored user databases (for example, SQL Server logins, associated server-level permissions, SQL Server Agent jobs).   These are the same objects you may have already scripted and copied over to the primary data center FCI. | Yes |  |  |
| 1. Ensure possible owners of the two FCIs set correctly, i.e., possible owners for INST\_A should be PRIMARYNODE1, PRIMARYNODE2; and the possible owners for INST\_B should be DRNODE1, DRNODE2. |  |  |  |
| 1. Create an availability group (this step involves both the primary and DR FCIs). You can set the availability mode to asynchronous or synchronous, depending upon the workload and network characteristics of your environment. Select manual failover for the availability groups. In an FCI+AG solution, the FCI failover is automatic, and the availability group failover is manual. For more information about how to configure failover for this solution, see [Creation and Configuration of Availability Groups](http://msdn.microsoft.com/library/ff878265.aspx). | Yes |  | Yes—to ensure that the endpoint ports are open and troubleshooting, as needed |
| 1. Create the availability group listener. This step is not needed if you have already configured this as part of creating the availability group. You can create the availability group listener by using Transact-SQL, SQL Server PowerShell, or a wizard in SQL Server Management Studio. For more information about using the various methods, see [Create or Configure an Availability Group Listener](http://msdn.microsoft.com/library/hh213080(v=sql.110).aspx). | Yes | Yes | Yes–to coordinate IP address and port considerations |

Table 2: Building the FCI+AG Solution in the disaster recovery data center

After you have finished these steps, in Windows Failover Cluster Manager you can see that a new group under Services and Applications is created with the same name as the availability group. Within that new group you’ll also find the availability group listener resource and associated listener IP addresses (see Figure 5).

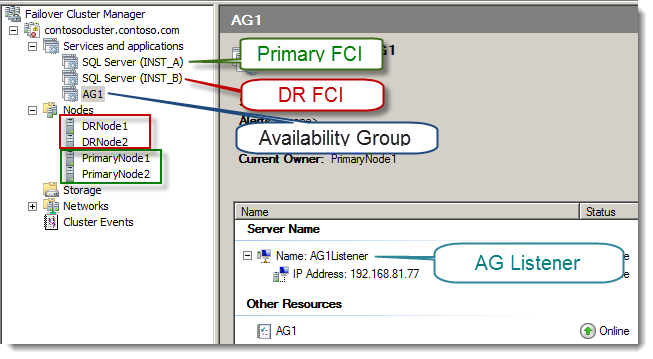


Figure 7: After configuration of the FCI for HA and AG DR design solution

Figure 7 shows the WSFC view of the deployment. Note that AG Listener in the figure shows one associated IP address for illustrative purposes; however, two IP addresses are common for multi-data center topologies.

**Note:** While the availability group appears as a resource in the WSFC, you should not attempt to manage it with Failover Cluster Manager or WSFC-scoped interfaces. Instead, manage the availability group within the context of the SQL Server instance via SQL Server Management Studio, Transact-SQL, or Windows PowerShell. For more information about why you should not use Failover Cluster Manager or other WSFC-scoped interfaces, see the blog post [DO NOT use Windows Failover Cluster Manager to perform Availability Group Failover](http://go.microsoft.com/fwlink/?LinkId=255057).

Figure 8 shows the deployment in SQL Server Management Studio. The view shows one of the FCIs and with the AlwaysOn High Availability Object Explorer folder hierarchy open. In this example, the DR FCI is the secondary replica and the other FCI is the primary replica. The three availability databases that participate in the group are listed, along with the name of the availability group listener.

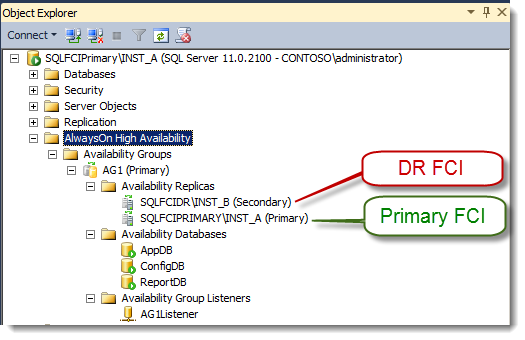


Figure 8: Post-configuration of the FCI for HA and AG DR design solution in SQL Server Management Studio

# Monitoring Considerations

Migrating from an FCI and database mirroring topology to an FCI and availability group solution will require new methods for monitoring the topology. The methods and tools you can use for monitoring the availability group infrastructure include the AlwaysOn Dashboard in SQL Server Management Studio, Object Explorer state information, Policy Based Management policies, new availability group related performance counters, catalog views, dynamic management views, and an Extended Events session that tracks recent AlwaysOn DDL related statement executions, WSFC connectivity issues, failover events, state changes, and redo thread blocks.

The AlwaysOn Dashboard is a recommended way to quickly view the health of a specific availability group. In it you can see the location of the primary instance, the failover mode of the replicas, the synchronization state of the replicas, and the failover readiness of the various replicas. You can also access the AlwaysOn Health Events Extended Events session data directly from the dashboard in order to view recent availability group activity, state changes, and events.

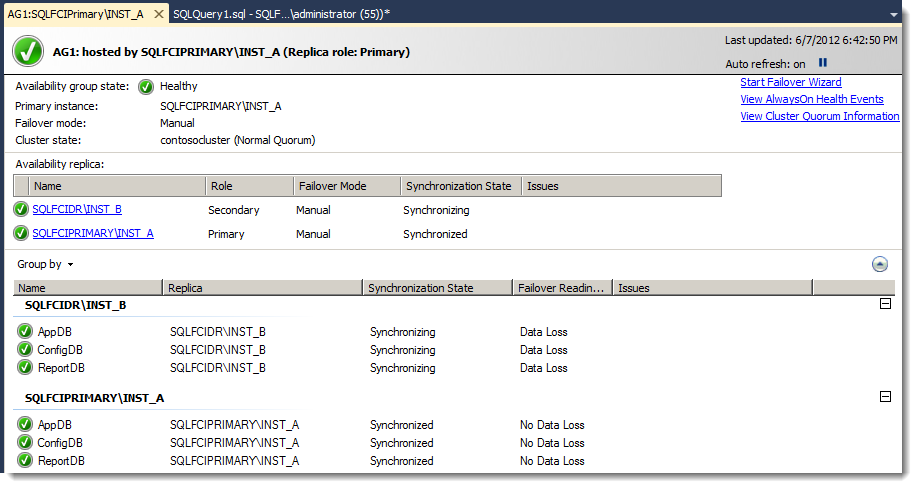


Figure 9: AlwaysOn Dashboard

Additionally you can create SQL Server Agent alerts and job responses based on performance counter thresholds and availability group state changes. For more information and guidance regarding the monitoring of an availability group environment, see [Monitoring of Availability Groups](http://msdn.microsoft.com/library/ff877954(v=sql.110).aspx).

# Recovering from a Disaster

This section details the workflow of steps you should take in the event of an outage of the primary replica in the primary data center. It also covers the steps that are needed to restore primary replica availability from the disaster recovery data center. An outage of the primary replica can be caused by one or more of the following:

* Failure of all the primary data center FCI nodes
* Failure of the primary data center FCI storage
* Failure or network outages impacting the entire primary data center

In any of these scenarios, certain actions are needed at the disaster recovery data center to resume SQL Server service to the applications.

Figure 10 shows the Cluster Quorum Information window for this scenario (this information is accessible from the AlwaysOn Dashboard and the **View Cluster Quorum Information** link). It shows the quorum before a disaster, where both DR nodes have zero votes.

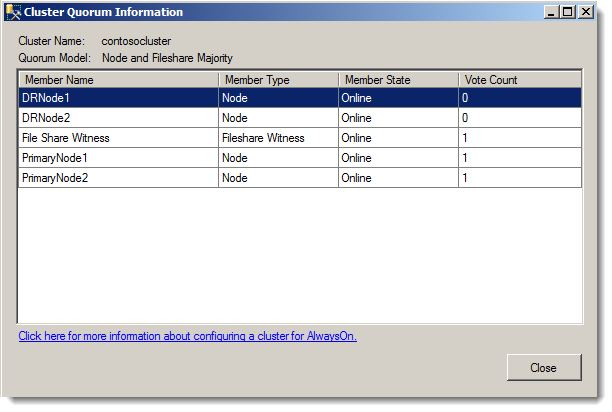


Figure 10: “Before” state of cluster quorum votes

The following workflow specifies the steps needed to recover an availability group in the disaster recovery data center in the event of a primary data center outage:

1. Force quorum on one of the DR nodes, and ensure that the nodes in the primary data center do not form their own quorum.

Failover Cluster Manager launched on a disaster recovery node is not likely to provide useful information (initially) on the state of the WSFC because the cluster no longer has quorum.



Figure 11: Failover Cluster Manager after a disaster and before recovery

Because the FCIs are dependent on a functioning WSFC, they are accessible unless both a cluster quorum and the cluster service are running. For a scenario where the primary data center’s status is uncertain and service must be restored from the DR secondary data center in order to conform to business recovery time objectives, you need to force quorum on one of the DR nodes. .

The following Windows PowerShell command demonstrates how to force quorum on one of the DR nodes.

Start-ClusterNode –Name "DRNODE1" –FixQuorum

After you execute this command, you should see something similar to the following.

Name State

------- --------

drnode1 Joining

**Note**: If the cluster service is still running on “DRNODE1”, you can use the following command in Windows PowerShell to stop the service before you start the cluster service again with force quorum::

Stop-ClusterNode –Name "DRNODE1"

For additional tools you can use to force quorum, such as cluster.exe or Failover Cluster Manager, see [Force a WSFC Cluster to Start Without a Quorum](http://msdn.microsoft.com/library/hh270275(v=sql.110).aspx).

1. Open Failover Cluster Manager to see the status of the Windows cluster. At this point, the Windows cluster should be up in the forced quorum state, and the secondary FCI should be up. The primary data center FCI will still be offline, as will the availability group resources.

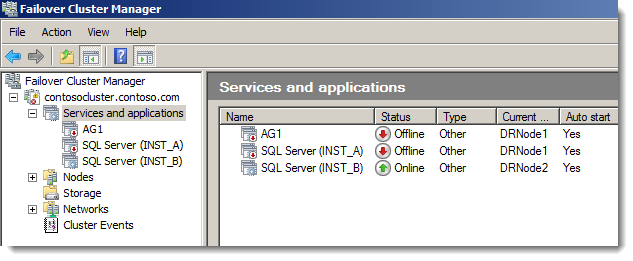


Figure 12: Failover Cluster Manager after forcing quorum

1. Bring the availability group online on the disaster recovery FCI.

**Caution:** If the replica was configured with asynchronous mode, *this means that restoring service could potentially result in data loss for any unsent log records*. Be sure to fully understand the consequences of this action.

For more information about what to do before, during, and after this type of manual failover, see [Perform a Forced Manual Failover of an Availability Group](http://msdn.microsoft.com/library/ff877957(v=sql.110).aspx).

Connect to the FCI in the DR data center using SQL Server Management Studio. SQL Server Management Studio should show the availability databases in a “not synchronizing” state. The DR FCI will also show as “resolving” as shown in Figure 13.

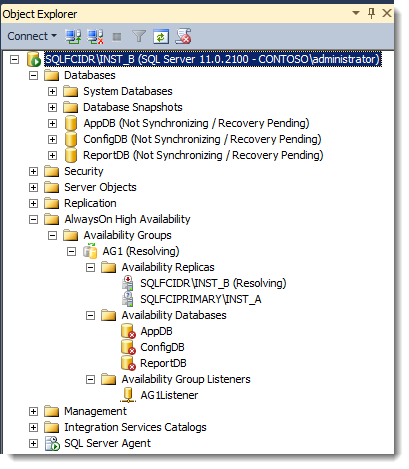


Figure 13: SQL Server Management Studio Object Explorer after forcing quorum

Note in Figure 13 that the other replica, which in this example is “SQLFCIPrimary\INST\_A” does not show any state in object explorer under the AG1 ‘Availability Replicas’ folder. This is the primary data center FCI that is no longer accessible due to the outage.

If the risk of some data loss is acceptable and service needs to be restored at the data center, execute the following Transact-SQL syntax at the disaster recovery FCI in order to force failover.

ALTER AVAILABILITY GROUP [AG1] FORCE\_FAILOVER\_ALLOW\_DATA\_LOSS;

At this point, your databases within the availability group should now be available. See Figure 14 for an example of the post-force failover state.

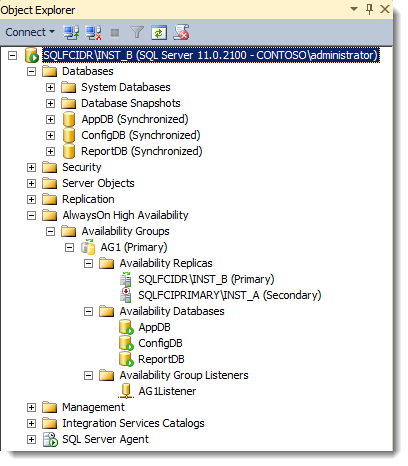


Figure 14: Object Explorer after failover is forced

After it comes back online, new connections to the availability group listener route automatically to the current primary replica, which is now hosted by the disaster recovery FCI.

Also note that you will still see various warning messages about the primary data center nodes being unavailable in SQL Server Management Studio. Figure 15 shows an example of what this may look like.

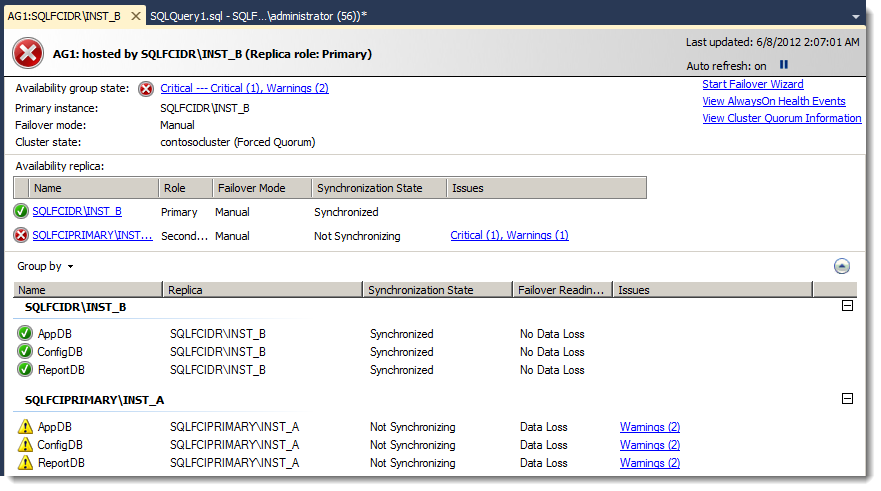


Figure 15: AlwaysOn Dashboard after a forced failover

1. From a DR WSFC node, remove votes from the primary data center nodes and give votes to DR data center nodes. Votes *can* be removed even though the primary data center nodes are not available. The two nodes assigned a weight of “1” are the DR WSFC nodes.

(Get-ClusterNode **"**DRNode1").NodeWeight=1

(Get-ClusterNode "DRNode2").NodeWeight=1

(Get-ClusterNode "PrimaryNode1").NodeWeight=0

(Get-ClusterNode "PrimaryNode2").NodeWeight=0

**Note:** If the DR site needs to be used for a longer period of time, it is recommended that additional voting members (WSFC node or file share) be added.

Before continuing, you can validate that the node votes were modified as intended by using the following Windows PowerShell command.

Get-ClusterNode | fl NodeName, NodeWeight

As mentioned earlier in the paper, large enterprise environments typically have a separation of duties among the database administrator, Windows Server (or cluster) administrator, and network administrator roles. The following table recaps the previously described disaster recovery workflow, indicating which areas typically fall under the various enterprise roles from a planning perspective.

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Database administrator | Windows Server \ cluster administrator | Network administrator |
| 1. Confirm the current state of the primary data center and the remaining WSFC disaster recovery nodes, coordinating efforts. | Yes | Yes | Yes |
| 1. Force quorum on one of the nodes at the DR site in order to access the DR FCI. |  | Yes |  |
| 1. Force failover of the availability group to the disaster recovery FCI. | Yes |  |  |
| 1. Add votes to the DR nodes and remove votes from the primary nodes. |  | Yes |  |

Table 3: Recovering from a disaster by job role

# Reverting Back to the Primary Data Center

With service restored at the disaster recovery data center, this scenario illustrated in this paper is assumed to be a temporary state until the primary data center issues are resolved. An outage scenario can have several variations and thus variations on recovery. The scenario described here assumes a disaster scenario where the primary data center servers are down for an extended period of time.

After the issues with the primary data center are resolved, and the nodes in the primary data center are powered on again, the nodes attempt to connect to the WSFC. After they are reconnected to the WSFC with cluster services running, the node weights assigned at the disaster recovery node should be in effect. This scenario also assumes that the original SQL Server installations and associated databases are still intact.

The replica on the previously failed primary data center FCI will be in a “not synchronizing” state (see Figure 16).

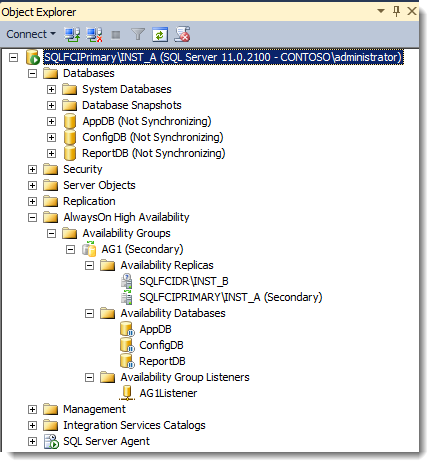


Figure 16: SQL Server Management Studio after primary FCI recovery but before resuming the availability group

The DR site SQL Server instance (in our example “SQLFCIDR\DC2”) is still the primary replica. Also notice the “pause” symbol by each availability database under the Availability Databases folder.

At this point you should evaluate whether you need to salvage data (that is, the data changes that were made in the original primary replica, but were unsent to the DR replica just prior to the disaster), or move forward instead with reestablishing the replica sessions.

**Caution:** Resuming the availability group replicas at this point may cause data loss, so if data loss is not acceptable, the data must be salvaged before data movement is resumed. Conversely, *not* resuming the availability group causes the transaction log files to keep growing on the DR replica databases.

One method to do this would be to create a database snapshot on the suspended secondary databases (original primary) for the purpose of extracting the data needed in order to resynchronize with the DR replica version of the availability databases. The following example demonstrates how to create a database snapshot on a “not synchronizing” availability database.

-- Create the database snapshot

CREATE DATABASE AppDB\_A1 ON

( NAME = AppDB, FILENAME =

'R:\MSSQL11.MSSQLSERVER\MSSQL\Data\AppDB\_A1.ss' )

AS SNAPSHOT OF AppDB;

GO

The required data can now be extracted from the database snapshot and inserted into the current primary replica appropriately, before moving forward with resuming data movement.

**Note**: For more information about the risks of forcing failover and mitigating data loss, see [Failover and Failover Modes](http://technet.microsoft.com/library/hh213151.aspx).

After the question of data loss is addressed appropriately, and it is time to revert service back to the primary data center, the next step is to move the primary replica role back to the primary data center in a controlled fashion:

1. Start the controlled migration back to the primary data center by adding back the quorum votes to the two primary data center nodes. After configuring this setting, be sure to verify again that all nodes in the WSFC have a vote.
2. To resume each database that participates in the availability group, execute Transact-SQL ALTER DATABASE commands on the primary data center FCI. Here is an example.

ALTER DATABASE AppDB SET HADR RESUME;

GO

ALTER DATABASE ConfigDB SET HADR RESUME;

GO

ALTER DATABASE ReportDB SET HADR RESUME;

GO

1. To synchronize prior to failover, modify the availability group on the DR FCI to temporarily use the synchronous commit availability mode. Ideally the synchronous commit setting should be made during a period of low application activity in order to minimize the impact of transaction latency on users.

Here is an example of the Transact-SQL command (executed on the current primary FCI in the disaster recovery data center), In this example, AG1 is the availability group, and the primary data center replica is designated as SQLFCIPrimary\INST\_A.

USE [master]

GO

ALTER AVAILABILITY GROUP [AG1]

MODIFY REPLICA ON N'SQLFCIPrimary\INST\_A' WITH (AVAILABILITY\_MODE = SYNCHRONOUS\_COMMIT);

GO

In the same SQL Server Management Studio session, execute the following command to set synchronous commit on the DR replica as well.

USE [master]

GO

ALTER AVAILABILITY GROUP [AG1]

MODIFY REPLICA ON N'SQLFCIDR\INST\_B' WITH

(AVAILABILITY\_MODE = SYNCHRONOUS\_COMMIT);

GO

1. Confirm the synchronization status between the two locations (both replica states should say “healthy” before moving to the next step, meaning that both synchronous-commit replicas are synchronized).

SELECT role\_desc,

synchronization\_health\_desc

FROM sys.dm\_hadr\_availability\_replica\_states;

1. To fail over from the disaster recovery data center FCI to the former primary data center FCI, connect and execute the following script on the primary data center FCI, which will become the new primary replica.

ALTER AVAILABILITY GROUP [AG1] FAILOVER;

1. If your topology uses high-performance mode, as mentioned earlier, change the disaster recovery FCI replica node back to asynchronous commit. Execute the following Transact-SQL on the primary replica.

USE [master]

GO

ALTER AVAILABILITY GROUP [AG1]

MODIFY REPLICA ON N'SQLFCIDR\INST\_B' WITH

(AVAILABILITY\_MODE = ASYNCHRONOUS\_COMMIT);

GO

USE [master]

GO

ALTER AVAILABILITY GROUP [AG1]

MODIFY REPLICA ON N'SQLFCIPrimary\INST\_A' WITH

(AVAILABILITY\_MODE = ASYNCHRONOUS\_COMMIT);

GO

1. Remove quorum votes from the disaster recovery nodes.

The following table recaps the previously described disaster recovery workflow, indicating which areas typically fall under the various enterprise roles from a planning perspective.

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Database administrator | Windows Server \ cluster administrator | Network administrator |
| 1. After primary data center service, nodes, and FCI are restored, add back quorum votes to the original primary nodes. |  | Yes |  |
| 1. Resume the availability database sessions on each secondary replica. | Yes |  |  |
| 1. Change the disaster recovery FCI replica and primary data center FCI replica to synchronous commit. | Yes |  |  |
| 1. Confirm the synchronization status between the two locations (both replica states should say “healthy” before moving to the next step). | Yes |  |  |
| 1. Fail over to the primary data center FCI replica. | Yes |  |  |
| 1. Revert the disaster recovery replica back to asynchronous commit (to match original configuration). | Yes |  |  |
| 1. Remove votes from the DR WSFC nodes. |  | Yes |  |

Table 4: Reverting to the primary data center

# Conclusion

SQL Server 2012 AlwaysOn Availability Groups can be used to replace database mirroring in topologies using FCIs for high availability and database mirroring for disaster recovery. This design pattern extends the capabilities beyond what was offered in earlier versions, allowing for a multi-database unit of failover, read-only replicas, and more. The intent of this white paper was to present a new HA and DR solution using AlwaysOn FCIs and AlwaysOn Availability Groups to replace the legacy architecture.

Successful deployment of such an HA/DR solution involves not just the DBA team, but close collaboration between the DBA team, Windows Server administration team, and the networking team in the IT organization. Cross-education of skills is very valuable when you deploy the HA/DR solution.

# References

* SQL Server 2012 AlwaysOn High Availability and Disaster Recovery Design Patterns (<http://go.microsoft.com/fwlink/?LinkId=255048>)
* Microsoft SQL Server AlwaysOn Solutions Guide for High Availability and Disaster Recovery (<http://msdn.microsoft.com/library/hh781257.aspx>)
* AlwaysOn Failover Cluster Instances (<http://technet.microsoft.com/library/ms189134.aspx>)
* Overview of AlwaysOn Availability Groups (<http://technet.microsoft.com/library/ff877884(v=SQL.110).aspx>)
* Failover Clustering and AlwaysOn Availability Groups (<http://technet.microsoft.com/library/ff929171.aspx>)
* Prerequisites, Restrictions, and Recommendations for AlwaysOn Availability Groups (<http://technet.microsoft.com/library/ff878487(v=sql.110).aspx>)
* Failover Cluster Step-by-Step Guide: Configuring the Quorum in a Failover Cluster (<http://technet.microsoft.com/library/cc770620(v=WS.10).aspx>)
* Windows Server hotfix for quorum votes (<http://support.microsoft.com/kb/2494036>)
* Windows PowerShell (<http://technet.microsoft.com/library/bb978526>)
* Mapping Cluster.exe Commands to Windows PowerShell Cmdlets for Failover Clusters (<http://technet.microsoft.com/library/ee619744(v=WS.10).aspx>)
* Windows PowerShell Survival Guide (<http://social.technet.microsoft.com/wiki/contents/articles/183.windows-powershell-survival-guide-en-us.aspx>)
* Failover Cluster Cmdlets in Windows PowerShell (<http://technet.microsoft.com/library/ee461009.aspx>)
* SQL Server PowerShell (<http://msdn.microsoft.com/en-us/library/hh245198.aspx>)

**For more information:**

SQL Server Web site (<http://www.microsoft.com/sqlserver/>)

SQL Server TechCenter (<http://technet.microsoft.com/sqlserver/>)

SQL Server DevCenter (<http://msdn.microsoft.com/sqlserver/>)

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

* Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?
* Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release.

[Send feedback](mailto:sqlfback@microsoft.com?subject=White%20Paper%20Feedback:%20AlwaysOn%20Architecture%20Guide:%20Building%20a%20High%20Availability%20and%20Disaster%20Recovery%20Solution%20by%20Using%20Failover%20Cluster%20Instances%20and%20Availability%20Groups)