

Migration Guide: Migrating to SQL Server 2012 Failover Clustering and Availability Groups from Prior Clustering and Mirroring Deployments

Part I – Prescriptive Guidance

**Writer:** Cephas Lin (Microsoft)

**Contributors:** Sanjay Mishra (Microsoft), David Smith (ServiceU Corporation), LeRoy Tuttle Jr. (Microsoft)

**Technical Reviewers:** Prem Mehra (Microsoft), Mike Weiner (Microsoft), Steve Howard (Microsoft), Kun Cheng (Microsoft), Paul Randall (SQLSkills)

**Published:** March 2012

**Applies to:** Microsoft SQL Server 2012

**Summary:** This paper provides guidance for customers who prior to SQL Server 2012 have deployed SQL Failover Clustering for local high availability and database mirroring for disaster recovery, and who want to migrate to SQL Server AlwaysOn. It describes the corresponding SQL Server AlwaysOn scenario and the migration paths to SQL Server AlwaysOn. It also contains the important knowledge and considerations that you must know in order to successfully migrate to a HADR solution based on SQL Server AlwaysOn technology, which implements AlwaysOn Failover Cluster Instances for high availability and AlwaysOn Availability Groups for disaster recovery.

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](#_Toc321126883)

[Migration Path to SQL Server 2012 AlwaysOn 4](#_Toc321126884)

[Migration Without Intermediate Hardware 5](#_Toc321126885)

[Migration with Intermediate Hardware 7](#_Toc321126886)

[Important Considerations 10](#_Toc321126887)

[AlwaysOn Availability Groups and Windows Server Failover Clustering 11](#_Toc321126888)

[Forced Quorum in AlwaysOn Availability Groups 11](#_Toc321126889)

[Automatic Failovers and FCI 12](#_Toc321126890)

[Client Connectivity Strategy During Migration 12](#_Toc321126891)

[Client Connectivity in Migration Without Intermediate Hardware 13](#_Toc321126892)

[Client Connectivity in Migration with Intermediate Hardware 13](#_Toc321126893)

[Conclusion 14](#_Toc321126894)

# Introduction

Many Tier-1 database solutions implement SQL Server Failover Clustering for high availability and database mirroring for disaster recovery. As Part I of a two-part series, this paper attempts to give you comprehensive guidance on what the SQL Server 2012 implementation looks like and how to get there. It describes the common migration paths and points out the important considerations you must have for a successful adoption of high availability and disaster recovery (HADR) with SQL Server 2012. Part II (http://msdn.microsoft.com/en-us/library/hh921498) of this series will provide an end-to-end walkthrough of an actual migration scenario.

This paper provides the “How” rather than the “Why” of migration to SQL Server AlwaysOn. For more information about the “Why”, including the features and capabilities of SQL Server AlwaysOn, see [Microsoft SQL Server AlwaysOn Solutions Guide for High Availability and Disaster Recovery](http://msdn.microsoft.com/en-us/library/hh781257.aspx) (http://msdn.microsoft.com/en-us/library/hh781257.aspx).

**Note**: SQL Server Failover Clustering has been renamed AlwaysOn Failover Cluster Instances (FCI) for SQL Server 2012. AlwaysOn FCI contains features and improvements that can be leveraged for faster performance, higher up time, reliable failure detection and diagnostics, predictable failovers, and much better resource utilization. For simplicity, the term *FCI* in this paper applies to both SQL Server Failover Clustering (prior to SQL Server 2012) and AlwaysOn FCI in (SQL Server 2012).

# Migration Path to SQL Server 2012 AlwaysOn

This paper focuses on the following high availability and disaster recovery scenario, in which FCIs are deployed for local high availability and database mirroring is deployed for remote disaster recovery. This configuration is the “Point A” in the migration scenario this paper discusses. Figure 1 shows the deployment architecture.



Figure A Typical HADR deployment involving FCIs and database mirroring

The corresponding solution in SQL Server 2012 is to deploy AlwaysOn FCIs for local high availability and AlwaysOn Availability Groups for disaster recovery. This is the “Point B” in the migration scenario. The post-migration architecture is shown in Figure 2.



Figure 2 HADR deployment in SQL Server 2012 involving FCIs and availability groups

There are two main approaches to migrate to an FCI and availability group solution in SQL Server 2012. With either approach, you should plan your migration steps to minimize application down time and to safeguard the integrity of your data. The pros and cons of each option are shown in the following table.

|  |  |
| --- | --- |
| Migration without intermediate hardware | Migration with intermediate hardware |
| * Simpler * No redundancy or data protection at various points during migration * Requires client reconfiguration during migration | * Involves more steps * Preserves high availability and/or disaster recovery capabilities during migration * Requires client reconfiguration during migration |

Table Main migration paths from FCIs and mirroring to SQL Server AlwaysOn

## Migration Without Intermediate Hardware

Migration to SQL Server AlwaysOn without intermediate hardware is the simpler path, and as the title implies, it uses fewer resources, but at the loss of redundancy and data protection during migration. It does not mean that your system will suffer significant down time, but your active nodes may become the single point of failure at various points of the migration sequence.

The general outline of the migration sequence is as follows:

1. Upgrade all secondary FCI nodes on the primary site (FCI1), as shown in Figure 3.



Figure Upgrade the secondary nodes on the primary site

1. Fail over to one of the upgraded nodes in FCI1, and then upgrade the previous primary node in FCI1, as shown in Figure 4. The databases are automatically upgraded to SQL Server 2012 when the upgraded instance is started at the end of the failover process. Client applications will experience a period of down time while your databases are being upgraded.

**Important**: After this step completes, you will not have data movement from the new SQL Server 2012 FCI to the remote FCI (FCI2), because FCI2 is running an older version of SQL Server. The difference in versions causes logs to be accumulated on FCI1, which cannot be truncated until data movement is resumed. To prevent your logs from filling up the log disk space, perform the next step immediately.



Figure Upgrade the remaining node on the primary site

1. Remove the database mirroring configuration and reconfigure FCI2 by joining the server nodes to the Windows Server Failover Clustering (WSFC) cluster on the primary site and reinstalling a new FCI2, as shown in Figure 5. You will need to completely remove the existing FCI2 because the underlying WSFC cluster must be destroyed in order for the nodes to be joined to the WSFC cluster for FCI1. It may be more efficient to completely reinstall Windows.

**Important**: Joining the server nodes to the WSFC cluster also means that you must reconsider your quorum model to maintain optimal WSFC quorum health both when and after adding the nodes. For more information, see the **WSFC Quorum Modes and Voting Configuration** section in [Microsoft SQL Server AlwaysOn Solutions Guide for High Availability and Disaster Recovery](http://msdn.microsoft.com/en-us/library/hh781257.aspx) (http://msdn.microsoft.com/en-us/library/hh781257.aspx).



Figure Reconfigure the remote site

**Note**: After you remove database mirroring, clients that connect with the Failover\_Partner parameter in the connection string can continue to connect to the databases as long as the Server parameter in the connection string points to the principal server in the database mirroring configuration (FCI1). For more information, see Client Connectivity Strategy During Migration.

1. Establish an availability group with an availability group listener, as shown in Figure 6. As when you configure database mirroring, you must back up and restore your databases to FCI2 in the same way before you create the new availability group. The New Availability Group Wizard contains an option to help automate this process, but it is not designed for very large databases that may exist in Tier-1 environments.



Figure Establish the availability group

**Note**: After the availability group is created, reconfigure your clients to connect to the availability group listener. For more information, see Client Connectivity Strategy During Migration.

## Migration with Intermediate Hardware

Migration to SQL Server AlwaysOn with intermediate hardware enables you to preserve high availability and disaster recovery capabilities during migration, but it involves more steps. This option is especially useful if you want to upgrade the operating systems and hardware components during migration.

For an end-to-end walkthrough of this migration path, see [Part II](http://msdn.microsoft.com/en-us/library/hh921498).

The general outline of the migration sequence is as follows:

1. Add a temporary FCI (FCI3 in Figure 7) to the primary site running SQL Server 2012. At this point, there are three WSFC clusters, two running in the primary site (hosting FCI1 and FCI3, respectively), and one running in the remote site (hosting FCI2). The newly created WSFC cluster that hosts FCI3 is highlighted in Figure 7 to show how it changes throughout the rest of the migration sequence. This new WSFC cluster will become the only WSFC cluster that remains at the end of the migration process.



Figure Add a Temporary FCI to the Primary Site

1. Break database mirroring with FCI2 and establish mirroring with FCI3, as shown in Figure 8. 

Figure Establish Database Mirroring with the Temporary FCI

**NOTE**: If the clients use the Failover\_Partner in the connection string, you must modify the Failover\_Partner parameter to point to FCI3 instead of FCI2. For more information, see Client Connectivity Strategy During Migration.

1. Reconfigure the FCI2 nodes with new hardware and software (optional), then join these nodes to the WSFC cluster for FCI3, and then install a SQL Server 2012 FCI on these nodes as FCI2. The new WSFC cluster (for FCI3) now contains two FCIs in two separate sites, as shown in Figure 9.

**Important**: Joining the server nodes to the WSFC cluster also means that you must reconsider your quorum model to maintain optimal WSFC quorum health both when and after adding the nodes. For more information, see the **WSFC Quorum Modes and Voting Configuration** section in [Microsoft SQL Server AlwaysOn Solutions Guide for High Availability and Disaster Recovery](http://msdn.microsoft.com/en-us/library/hh781257.aspx) (http://msdn.microsoft.com/en-us/library/hh781257.aspx). 

Figure Add the Remote FCI to the New WSFC Cluster

1. Perform a database mirroring failover in the primary site to make FCI3 the new primary partner, then remove the database mirroring configuration, and then establish an availability group with an availability group listener, with FCI3 as the primary replica and FCI2 as the secondary replica, as shown in Figure 10.



Figure Establish Availability Group Between Temporary and Remote FCIs and Reconfigure Primary FCI

**Note**: After the availability group is created, reconfigure your clients to connect to the availability group listener. Clients will experience a period of down time from the time that database mirroring is stopped until they are reconfigured to point to the new availability group listener. To increase client connectivity, you can modify your clients to point directly to FCI3 immediately after stopping database mirroring. For more information, see Client Connectivity Strategy During Migration.

1. Reconfigure the FCI1 nodes with new hardware and software (optional), then join these nodes to the new WSFC cluster (for FCI3), then configure a new SQL Server 2012 FCI on these nodes as FCI1, and then add FCI1 to the availability group as a secondary replica. The resulting configuration is shown in Figure 11.

**Important**: Joining the server nodes to the WSFC cluster also means that you must reconsider your quorum model to maintain optimal WSFC quorum health both when and after adding the nodes. For more information, see the **WSFC Quorum Modes and Voting Configuration** section in [Microsoft SQL Server AlwaysOn Solutions Guide for High Availability and Disaster Recovery](http://msdn.microsoft.com/en-us/library/hh781257.aspx) (http://msdn.microsoft.com/en-us/library/hh781257.aspx).



Figure Install the new primary FCI and add it to the availability group

1. Make FCI1 the new primary availability replica, and then remove FCI3 from the availability group and evict the corresponding nodes from the WSFC cluster, as shown in Figure 12.



Figure Remove the temporary FCI

# Important Considerations

You should be aware of some key conceptual differences between the SQL Server 2008 R2 solution and SQL Server AlwaysOn when you plan a migration to an FCI and availability group solution. Before you start the migration walkthrough, make sure you understand all the concepts outlined in this section.

## AlwaysOn Availability Groups and Windows Server Failover Clustering

That the Windows Server Failover Clustering (WSFC) service is integral to AlwaysOn Failover Cluster Instances is not new. When you implement FCIs for high availability and database mirroring for disaster recovery, you have two FCIs that reside in two WSFC clusters, one on the primary site and one on the remote site. When you use an availability group for disaster recovery, however, the availability group requires all replicas to be *in the same WSFC cluster*. You must pay attention to the following:

* **Windows Domain** Because the FCIs in both the primary and remote sites must run in the same WSFC cluster, all server nodes must also run in the same Windows domain.
* **Firewall Ports** To run a WSFC cluster in a multi-site environment, you must make sure that all respective firewall ports are open between the sites in order to allow the WSFC server to join nodes on the remote site and for the multi-site WSFC cluster to function properly.
* **FCI Naming** According to previous best practices, FCI names in the primary and remote sites should be the same. However, the requirement for the FCIs to be in the same WSFC cluster means that the FCI names cannot be the same.
* **Quorum Configuration** Because the WSFC cluster for the availability group now spans multiple sites, you must carefully plan your quorum configuration in order to preserve quorum health in the event of common problems such as network segmentation. You do not want your database to go down simply because your primary and remote sites lose connectivity with each other.

It is crucial that you understand the concept of WSFC quorum in order to be successful in SQL Server AlwaysOn. This is especially true for a multi-subnet solution like the one discussed in this paper. For more information, see [Microsoft SQL Server AlwaysOn Solutions Guide for High Availability and Disaster Recovery](http://msdn.microsoft.com/en-us/library/hh781257.aspx) (http://msdn.microsoft.com/en-us/library/hh781257.aspx).

* **WSFC Disaster Recovery** As part of your disaster recovery plan, ensure that you have a working plan to recover the WSFC quorum if it is lost. The quorum setting may need to be reconfigured multiple times during the recovery process. For more information about best practices for WSFC disaster recovery, see [Microsoft SQL Server AlwaysOn Solutions Guide for High Availability and Disaster Recovery](http://msdn.microsoft.com/en-us/library/hh781257.aspx) (http://msdn.microsoft.com/en-us/library/hh781257.aspx).
* **Cluster Resource Ownership** When the two FCIs are each hosted by a WSFC cluster, all nodes in each cluster are possible owners for all cluster resources. However, implementing two FCIs in the same WSFC cluster means each FCI resource must have a subset of nodes as possible owners instead of all the nodes in the WSFC cluster. By default, all WSFC nodes are assigned possible ownership of all FCI resource groups. You should exercise care in reassigning resource group ownership to the correct nodes in order to avoid failure during setup and during failovers.

## Forced Quorum in AlwaysOn Availability Groups

When you use asynchronous database mirroring in SQL Server 2008 R2, the mirror database on the remote site needs to perform a *forced service* to come online if a disaster causes the primary site to go offline. Similarly, in a typical scenario using AlwaysOn Availability Groups, the secondary replica in the remote site needs to perform a *forced quorum* and a *forced failover* to come online if a disaster causes the primary site to go offline, because this occurrence usually causes the WSFC cluster to lose quorum.

The key difference is that a forced service in database mirroring is performed at the database level, but a forced quorum is performed at the WSFC level and the forced failover is performed at the database level. The loss of the WSFC quorum health causes the WSFC cluster, along with its FCIs and any availability groups it hosts, to go offline as well. Therefore, you must first perform a forced quorum on the WSFC service directly and make sure that the FCI and availability groups are back online; then you must perform a forced failover on the availability group using a Transact-SQL command.

For more information, see:

* [Disaster Recovery (AlwaysOn Availability Groups)](http://msdn.microsoft.com/en-us/library/hh403411(SQL.110).aspx) (http://msdn.microsoft.com/en-us/library/hh403411(SQL.110).aspx)
* [Force a WSFC Cluster to Start Without a Quorum](http://msdn.microsoft.com/en-us/library/hh270275(v=SQL.110).aspx) (http://msdn.microsoft.com/en-us/library/hh270275(v=SQL.110).aspx).
* [ALTER AVAILABILITY GROUP (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ff878601(v=SQL.110).aspx) (http://msdn.microsoft.com/en-us/library/ff878601(v=SQL.110).aspx)

## Automatic Failovers and FCI

When an FCI is joined to an availability group, the nodes within the FCI can fail over to each other automatically, but the availability replica on the FCI must manually fail over to or from another replica in the availability group. AlwaysOn Availability Groups does not support automatic failovers to and from an FCI. This should not be a cause for concern because you generally do not want automatic failovers to a disaster recovery site, because you do not want an automatic failover to happen simply because your primary and remote sites lose connectivity with each other.

## Client Connectivity Strategy During Migration

Depending on your specific migration scenario, your client connection strings may need to be changed multiple times in order to maintain connectivity during the migration process. Connection strings that client applications use to connect to the premigration solution may look similar to the following:

Server=*<Partner\_A>*\*<Port>*; Failover\_Partner=*<Partner\_B>*\*<Port>*; Database=*<DB\_Name>*; Network=dbmssocn

In certain steps of the migration sequence, you can simply disconnect the mirror server in the database mirroring configuration (change mirroring state to *disconnected*) to allow this connection string to work. However, this approach is not recommended for very high transaction volume workloads. When you disconnect the mirror server, the log disk space on the principal server continues to fill up until the database mirroring session is reestablished. For Tier-1 implementations with high transaction throughput numbers, this is not a viable option, given that the duration for reconfiguring new FCIs and reestablishing a mirroring session can take a long time, and the principal server is likely to suffer from disk space exhaustion.

As an alternative, consider removing the database mirroring configuration altogether instead. The connection string continues to work if the Server parameter points to the principal server in the database mirroring configuration that has since been removed. This approach does not cause the transactional logs to fill up.

In addition, AlwaysOn Availability Groups are designed to be backward compatible with database mirroring, so the Server=*<Partner\_A>*\*<Port>*; Failover\_Partner=*<Partner\_B>*\*<Port>*; construct can continue to work if the following are true:

* The availability group only contains one nonreadable secondary replica (analogous to the mirror server).
* The FCI names do not need to be modified (see AlwaysOn Availability Groups and Windows Server Failover Clustering).

Doing so makes it possible to tweak the steps of Migration Without Intermediate Hardware to maintain client connectivity without the need for any client reconfiguration.

### Client Connectivity in Migration Without Intermediate Hardware

As discussed earlier, it is possible to maintain client connectivity without client reconfiguration in Migration Without Intermediate Hardware. Consider the following approach:

1. Before you start the migration process, make sure that all client connection strings have FCI1 on the primary site in the Server parameter and (if using the Failover\_Partner parameter) FCI2 on the remote site in the Failover\_Partner parameter.
2. When you create the availability group, configure FCI2 on the remote site as a non-readable secondary replica.
3. At your convenience, modify all client connection strings by removing the Failover\_Partner parameter and using the availability group listener in the Server parameter.
4. After all client connection strings have been modified, reconfigure FCI2 on the remote site as a readable secondary replica, if desired.

### Client Connectivity in Migration with Intermediate Hardware

To reduce the need for client reconfiguration in Migration With Intermediate Hardware, consider the following approach:

1. Create a VNN on the DNS server and point it to the principal server in the database mirroring configuration (FCI in the primary site). Give this VNN a name that is different from the availability group listener you plan to use.
2. Make sure both FCIs on the primary and remote sites, as well as the temporary FCI you create, listen to the same port. This makes it possible to redirect the VNN to the remote site if you must perform disaster recovery during the migration process.
3. Prior to the migration process, reconfigure your clients to use the following connection string:

Server=VNN-Principal\*<port>*; Database=*<DB\_Name>*; Network=dbmssocn

1. Whenever the database mirroring configuration changes, update the VNN to point to the new principal server.
2. Immediately after you create the availability group with the availability group listener, modify the VNN to point to the availability group listener.

For more information, see:

* [Availability Group Listeners, Client Connectivity, and Application Failover (SQL Server)](http://msdn.microsoft.com/en-us/library/hh213417(v=sql.110).aspx) (http://msdn.microsoft.com/en-us/library/hh213417(v=sql.110).aspx)
* [Microsoft SQL Server AlwaysOn Solutions Guide for High Availability and Disaster Recovery](http://msdn.microsoft.com/en-us/library/hh781257.aspx) (http://msdn.microsoft.com/en-us/library/hh781257.aspx)

# Conclusion

SQL Server 2012 offers new capabilities for HADR solutions with SQL Server AlwaysOn technology. This paper gives practical guidance for you to migrate from a FCI and database mirroring HADR solution prior to SQL Server 2012 to a SQL Server AlwaysOn solution in SQL Server 2012. It describes two main approaches. [Part II](http://msdn.microsoft.com/en-us/library/hh921498) (http://msdn.microsoft.com/en-us/library/hh921498) of this paper will walk you step-by-step through the second approach: Migration with Intermediate Hardware.

**For more information:**

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

<http://technet.microsoft.com/en-us/sqlserver/>: SQL Server TechCenter

<http://msdn.microsoft.com/en-us/sqlserver/>: SQL Server DevCenter

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:%20Migration%20Guide:%20Migrating%20to%20SQL%20Server%202012%20Failover%20Clustering%20and%20Availability%20Groups%20from%20Prior%20Clustering%20and%20Mirroring%20Deployments%20(Part%20I)).