

**Migration Guide: Migrating to AlwaysOn Availability Groups from Prior Deployments Combining Database Mirroring and Log Shipping**

**Part I – Prescriptive Guidance**

**Writer:** Cephas Lin (Microsoft)

**Contributors:** Sanjay Mishra (Microsoft)

**Technical Reviewers:** Aditya Samant (Microsoft), Juergen Thomas (Microsoft), Prem Mehra (Microsoft), Charles Mathews (Microsoft), Jimmy May (Microsoft), David P. Smith (ServiceU)

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

**Published:** August 2012

**Applies to:** Microsoft SQL Server 2012

**Summary:** This migration guide provides guidance for customers who have deployed database mirroring for local high availability and log shipping for disaster recovery based on SQL Server 2008 R2 or earlier and now want to upgrade to use SQL Server 2012 AlwaysOn Availability Groups. The migration sequence presented here is a best-practice approach that preserves the high availability and disaster recovery capabilities of your databases during most of the migration process.

This guide is Part I of a two-part series that describes planning considerations and deployment of a complete migration scenario. The information in this Migration Guide series can help you to successfully move to a high availability and disaster recovery (HADR) solution based on AlwaysOn Availability Groups.

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](#_Toc328347835)

[Migration Path to SQL Server 2012 AlwaysOn 4](#_Toc328347836)

[Migration Sequence 5](#_Toc328347837)

[Important Considerations 9](#_Toc328347838)

[AlwaysOn Availability Groups and Windows Server Failover Clustering 10](#_Toc328347839)

[Forced Quorum in AlwaysOn Availability Groups 10](#_Toc328347840)

[Client Connectivity Strategy During Migration 10](#_Toc328347841)

# Introduction

Many database solutions implement Microsoft SQL Server database mirroring for local high availability (HA) and log shipping for disaster recovery (DR). The SQL Server 2012 AlwaysOn Availability Groups solution provides significantly improved HA and DR capabilities compared to database mirroring and log shipping solutions that are based on previous versions of SQL Server. Migration from a database mirroring and log shipping solution to AlwaysOn Availability Groups can be a complicated process, and you should carefully design a plan that minimizes both planned and unplanned downtime during and after migration. This paper is Part I of a two-part series that describes planning considerations and deployment of a complete migration scenario.

Part I provides comprehensive guidance on deploying a SQL Server 2012–based solution. It describes the common migration paths and points out important considerations necessary for a successful adoption of high availability and disaster recovery (HADR) with SQL Server 2012.

[Part II](http://msdn.microsoft.com/en-us/library/jj635219) (http://msdn.microsoft.com/en-us/library/jj635219) of this series provides an end-to-end walkthrough of an actual migration scenario. It is intended to provide a prototype for you to use as the basis for developing your own migration deployment. Part II provides a proof-of-concept study for the considerations introduced in Part I.

This paper is intended to provide the “How” rather than the “Why” of migration to SQL Server AlwaysOn. For 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/library/hh781257.aspx) (http://msdn.microsoft.com/library/hh781257.aspx).

This paper also limits its discussion to standalone SQL Server instances and does not include AlwaysOn Failover Cluster Instances. For more information about migration with AlwaysOn Failover Cluster Instances, see [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) (http://msdn.microsoft.com/library/hh923056.aspx).

# Migration Path to SQL Server 2012 AlwaysOn

This paper focuses on the following high availability and disaster recovery scenario, in which database mirroring is deployed for local high availability and log shipping is deployed for remote disaster recovery. This configuration is the “Point A” of the migration scenario. The deployment architecture is shown in Figure 1.



Figure 1 A typical HADR deployment that uses database mirroring and log shipping

The corresponding solution in SQL Server 2012 is to deploy AlwaysOn Availability Groups for both high availability and disaster recovery. This is the “Point B” of the migration scenario. The post-migration architecture is shown in Figure 2. This scenario requires no new hardware.



Figure 2 HADR deployment in SQL Server 2012 that uses AlwaysOn Availability Groups

# Migration Sequence

The migration sequence in this section is a best-practice approach that preserves the HADR capabilities of your databases during the migration process. It is designed for a migration without any changes to existing hardware. If your solution requires changes to existing hardware, you must modify the sequence to optimize HADR capabilities and client connectivity.

In brief, the migration sequence can be summarized into four stages:

1. Online upgrade of the log shipping secondary database server (LS2ND).
2. Online upgrade of the mirror database server (DBM2).
3. Online upgrade of the principal database server (DBM1).
4. Configuration of an availability group.

For a list of prerequisites for implementing AlwaysOn Availability Groups on your servers, see [Server Instance Prerequisites and Restrictions](http://msdn.microsoft.com/en-us/library/ff878487.aspx#ServerInstance) (http://msdn.microsoft.com/en-us/library/ff878487.aspx#ServerInstance). For more information about the migration process, see Important Considerations.

A detailed step-by-step migration walkthrough is covered in [Migration Guide: Migrating to AlwaysOn Availability Groups from Prior Deployments Combining Database Mirroring and Log Shipping, Part II](http://msdn.microsoft.com/en-us/library/jj635219) (http://msdn.microsoft.com/en-us/library/jj635219). The general outline of the migration sequence is as follows:

1. Make sure that the restore job on LS2ND is set to WITH NORECOVERY. If the job is configured WITH STANDBY, the restore job will not run after you upgrade to SQL Server 2012.
2. Upgrade LS2ND to SQL Server 2012, as highlighted in Figure 3.

While LS2ND is being upgraded, DBM1 continues to ship logs, but the copy and restore jobs do not run on the secondary database, causing unrestored transaction log backups to accumulate. After LS2ND has been upgraded, the log shipping agents jobs resume and continue to copy and restore log backups from the primary database.



Figure 3 Upgrade the log shipping secondary database server

1. Upgrade the witness server, as shown in Figure 4.

Because the witness server is not needed for the availability group, you can simply remove it from the database mirroring configuration altogether. However, if you plan to perform migration in stages over a period of days, it is recommended that you keep the witness server in the database mirroring configuration for as long as possible. It helps preserve the HADR capabilities of your SQL Server system until you actually transition to SQL Server AlwaysOn.



Figure 4 Upgrade the witness server

1. Upgrade DBM2 to SQL Server 2012, as shown in Figure 5.

After the upgrade completes, data movement continues for both your database mirroring and log shipping configuration. Your databases are highly available in this configuration because they can fail over automatically from DBM1 to DBM2 if a failure occurs. After the databases fail over to DBM2, you no longer have high availability because it is not possible to synchronize data with, or to fail over to, an earlier version of SQL Server (DBM1).



Figure 5 Upgrade the mirror database server

1. Perform a database mirroring failover from DBM1 to DBM2, and then *immediately* upgrade DBM1, as shown in Figure 6.

In between the failover and the upgrade actions, the database mirroring session is suspended because DBM1 has an earlier version of SQL Server than DBM2, the new principal server. Because logs are accumulating on DBM2, you should upgrade DBM1 as rapidly as possible to resume data mirroring before storage is exhausted on DBM2.



Figure 6 Upgrade the original principal database server

**Important**: OLTP-workload clients that use the Failover\_Partner parameter in the connection string continue to connect to the database mirroring session. If some clients do not use the Failover\_Partner parameter in the connection string, make sure that they point to DBM2. For more information, see Client Connectivity Strategy During Migration.

1. Create a Windows Server Failover Clustering (WSFC) cluster that includes all three database servers, as shown in Figure 7. Before you join LS2ND to the WSFC cluster, make sure that all three servers on the primary and the remote sites are joined to the same Windows domain.

**Important**: Creating a WSFC cluster also means that you must carefully configure your quorum model to maintain optimal WSFC quorum health. 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/library/hh781257.aspx) (http://msdn.microsoft.com/library/hh781257.aspx). 

Figure 7 Create the WSFC cluster

1. Remove both the database mirroring and the log shipping configurations, and then create an availability group that encompasses all three database servers, which have now been joined to the WSFC cluster, as shown in Figure 8. When you create the availability group, be sure to configure an availability group listener as well. Configure all OLTP-workload clients to connect to the new availability group listener. For more information, see Client Connectivity Strategy During Migration.



Figure 8 Create the availability group

**Note**: Similar to database mirroring, creating the availability group requires you to prepare the databases by taking a full backup and a log backup and restoring them on the secondary replicas. In this migration process, however, this step is not required as long as DBM2 still holds the mirror copy and LS2ND has restored the log tail backup.

1. If you prefer to serve the databases from DBM1 (because it has better hardware, for example), you can now fail over the availability group to it.



Figure 9 Fail over the availability group at the primary site

# Important Considerations

There are key conceptual differences between the SQL Server 2008 R2 solution and AlwaysOn Availability Groups in SQL Server 2012 to be aware of when you migrate to an availability group solution. For more information about the concepts required for implementing an availability group solution, see [AlwaysOn Architecture Guide: Building a High Availability and Disaster Recovery Solution by Using AlwaysOn Availability Groups](http://technet.microsoft.com/library/jj191711.aspx) **(http://technet.microsoft.com/library/jj191711.aspx)**. Review and understand all the concepts in that white paper as well as the supplemental points described in this section before you start implementing your own migration scenario.

## AlwaysOn Availability Groups and Windows Server Failover Clustering

Windows Server Failover Clustering (WSFC) service is integral to AlwaysOn Availability Groups. While database mirroring and log shipping are managed directly in SQL Server do not have domain and clustering requirements, AlwaysOn Availability Groups require all replicas to be in the *same Windows domain* and the *same WSFC cluster*.

## Forced Quorum in AlwaysOn Availability Groups

Because AlwaysOn Availability Groups rely on a single WSFC cluster between the sites, the primary site may be offline during a disaster, causing the WSFC node on the remote site to be offline as well due to the loss of its WSFC quorum. In this scenario, the secondary replica in the remote site needs a *forced quorum* and a *forced failover* to come online. This scenario is analogous to a *forced service* in database mirroring without a witness server.

A forced service in database mirroring is performed at the database level. For AlwaysOn Availability Groups, however, 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 the availability groups it hosts, to go offline as well. Therefore, you must first force quorum on the WSFC service directly and make sure that the availability groups on the remote site (the secondary replicas) are back online. You can then force a failover on the availability groups on the remote site using a Transact-SQL command.

For more information, see:

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

* [Perform a Forced Manual Failover of an Availability Group](http://msdn.microsoft.com/library/ff877957(v=sql.110).aspx) (http://msdn.microsoft.com/library/ff877957(v=sql.110).aspx)

## Client Connectivity Strategy During Migration

Connection strings that client applications use to connect to the premigration solution look similar to the following.

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

This connection string continues to work during migration if one of the following conditions is true:

* Both the principal and mirror databases are online and the mirrored database fails over between the primary and mirror servers.
* The principal server is online, the Server parameter points to the principal server, and database mirroring is disconnected, suspended, or paused.
* An availability group is established in place of the original database mirroring configuration and hosts one primary replica and *only one* nonreadable secondary replica, and the replicas are used in the Server and Failover\_Partner parameters in the connection string.

In Step 7 of the migration sequence, after you remove database mirroring to create the availability group, your server configuration likely will not satisfy any of the conditions. To preserver client connectivity, you must modify the connection string by removing the Failover\_Partner parameter and pointing directly to the availability group listener or the primary replica.

If you need to separate client reconfiguration from the Migration Sequence, consider the following possible approach:

1. Create a CNAME record on the DNS server and point it to the principal/primary database server. Give this record a name that is different than the availability group listener you will use.
2. Make sure all database servers on the primary and remote sites listen to the same port. This makes it possible to redirect the CNAME record as necessary when you must perform an unplanned failover during the migration process.
3. Prior to the migration process, reconfigure your clients with the following connection string.

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

1. In Step 7 of the migration sequence, modify the CNAME record to point to the availability group listener immediately after you create the availability group.

For more information, see:

* [Availability Group Listeners, Client Connectivity, and Application Failover (SQL Server)](http://msdn.microsoft.com/library/hh213417(v=sql.110).aspx) (http://msdn.microsoft.com/library/hh213417(v=sql.110).aspx)
* [Microsoft SQL Server AlwaysOn Solutions Guide for High Availability and Disaster Recovery](http://msdn.microsoft.com/library/hh781257.aspx) (http://msdn.microsoft.com/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 database mirroring and log shipping HADR solution built on a version of SQL Server that is earlier than SQL Server 2012 to a SQL Server AlwaysOn solution in SQL Server 2012. [Migration Guide: Migrating to AlwaysOn Availability Groups from Prior Deployments Combining Database Mirroring and Log Shipping, Part II](http://msdn.microsoft.com/en-us/library/jj635219) (http://msdn.microsoft.com/en-us/library/jj635219) will walk you step-by-step through the migration steps.

**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)).