

# Database Mirroring and Log Shipping Working Together

SQL Server Best Practices Article

Writer: Sanjay Mishra

Contributors: Prem Mehra, Lindsey Allen

Technical Reviewers: Matt Hollingsworth, Juergen Thomas, Mark Souza, Burzin Patel, Tom Michaels, Raffaella Canobbio, Steve Lindell, Kevin Farlee, Mark Scurrell, Michael Redman, Michael Thomassy, Mike Ruthruff, Thomas Davidson

Published: January 2008

Applies To: SQL Server 2005 Service Pack 2

Summary: Log shipping and database mirroring can work together to provide solutions for high availability and disaster recovery. You can convert an existing log shipping configuration to a database mirroring configuration. Although Microsoft only supports a single mirror database, you can set up log shipping from a database mirroring pair to additional servers, to provide multiple failover sites. Moreover, you can easily switch which pair of instances is using database mirroring and which instances are configured with log shipping.

### Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

 2008 Microsoft Corporation. All rights reserved.

Microsoft and SQL Serverare either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

### Table of Contents

[Introduction 1](#_Toc187248326)

[Converting a Log Shipping Configuration to Database Mirroring 1](#_Toc187248327)

[Set Up 2](#_Toc187248328)

[Log Shipping to a Third Destination from a Database Mirroring Pair 4](#_Toc187248329)

[Set Up 5](#_Toc187248330)

[Potential Problem 6](#_Toc187248331)

[Switching Between the Database Mirroring Pair and the Log Shipping Pair 7](#_Toc187248332)

[Conclusion 8](#_Toc187248333)

#### Introduction

In this paper we discuss the following scenarios:

* Converting a log shipping configuration to database mirroring
* Log shipping to a third destination from a database mirroring pair
* Switching between the database mirroring pair and the log shipping pair

This paper assumes that the reader has a basic understanding of log shipping and database mirroring.

#### Converting a Log Shipping Configuration to Database Mirroring

Log shipping has been available for many years, and with the advent of database mirroring in Microsoft® SQL Server™ 2005 Service Pack 1 (SP1), it may be advantageous to convert log shipping installations to database mirroring. Database mirroring provides the following advantages over log shipping:

* No data loss option (synchronous database mirroring)
* Automatic failover option (synchronous database mirroring with witness)

Before converting from log shipping to database mirroring, note the following considerations:

* Database mirroring requires the FULL recovery model for the database, whereas log shipping can work with either the BULK\_LOGGED or FULL recovery models. If your database is running in the bulk-logged recovery model, you must change it to full to use database mirroring. Be aware that the FULL recovery model may generate a more transaction log.
* Log shipping has little or no impact on application performance on the primary server. Depending on the workload and the amount of the generated log, database mirroring can impact application performance. For details, see [Database Mirroring Best Practices and Performance Considerations](http://www.microsoft.com/technet/prodtechnol/sql/2005/technologies/dbm_best_pract.mspx). Evaluate your performance service level agreements with respect to your workload and the network capacity between the two servers.

You can convert an existing log shipping configuration to a database mirroring configuration without having to take a full backup of the principal database and restore it on the mirror. A simplified architecture for this deployment is shown in Figure 1.

****Figure 1: Converting a log shipping configuration to database mirroring****

##### Set Up

To convert a log shipping configuration to database mirroring

1. Ensure that log shipping is working fine between S1 and S2 – S1 as primary and S2 as secondary. You can check the status of the log shipping jobs (backup, copy and restore) from the SQL Agent job history. Make sure these jobs are running as scheduled without any errors. Also, ensure that the secondary database is in the NORECOVERY state, and not in the STANDBY state.
2. Set up database mirroring security, endpoints, permissions, and so on.
3. Disable the log backup job on the primary server. Wait until all the log backups taken so far have been copied and applied to the secondary server. You can verify this by checking the status of the restore job on the secondary server in the View History panel of the corresponding SQL Agent job in SQL Server Management Studio. Look for the file name of the last log backup restored on the secondary server. If it is the last log backup taken on the primary, the secondary has been rolled forward to the required point in time.

If you do not disable the log backup job on the primary, it will continue to create new log backups, and you may experience the following error while trying to set up database mirroring partnership:

The remote copy of database "<database name>" has not been rolled forward to a point in time that is encompassed in the local copy of the database log.

This error occurs when some transaction log backups have been taken on the primary server, but have not yet been copied and restored to the secondary server. Log shipping works on a schedule, and it is possible that at any given point in time, there may be some transaction log backups taken on the primary, but not yet restored to the secondary.

After you have disabled the log backup job, instead of waiting for the copy and restore jobs to execute on schedule, you can manually execute the copy job and the restore job on the secondary server. By default, the copy job and the restore job are scheduled to run every 15 minutes; they may have been scheduled to run even less frequently. Manually executing the copy job and the restore job reduces wait time.

1. After you ensure that all the log backups taken so far have been copied and applied to the secondary, disable the copy job and the log restore job on the secondary.
2. Enable database mirroring with S2 as the mirror and S1 as the principal. By default, database mirroring is configured as synchronous. If you desire asynchronous database mirroring between S1 and S2, you need to change safety level accordingly.
3. Remove the log shipping configuration between S1 and S2.

**Note**: Having both database mirroring and log shipping enabled to the same destination database at the same time is not supported. If database mirroring and log shipping are enabled to the same destination database, you will see errors similar to the following in the job history of the log restore job on the secondary:

Error: Could not apply log backup file 'D:\Backup\MyDB\_20070822224713.trn' to secondary database 'MyDB'.(Microsoft.SqlServer.Management.LogShipping)

Error: RESTORE cannot operate on database 'MyDB' because it is configured for database mirroring. Use ALTER DATABASE to remove mirroring if you intend to restore the database.

The above workflow describes how to easily convert an existing log shipping configuration to database mirroring.

This approach may be useful in another use case. Assume you have a log shipping configuration (without database mirroring) between two servers—S1 (the primary server) and S2 (the secondary server). Your business requirement is to minimize the downtime of the application for planned maintenance of the primary server. One common approach is to failover the database from the log shipping primary to the secondary (see [Failing Over to a Log Shipping Secondary](http://msdn2.microsoft.com/en-us/library/ms191233.aspx)), but the application will not fail over automatically and the downtime can be in minutes. An alternative approach to ensure a seamless user experience during planned downtime is as follows.

1. Convert the log shipping configuration to synchronous database mirroring by using the set up steps described earlier in this section. Depending on the workload and network capacity between the primary and secondary servers, you might want to pick a low activity window so that the mirroring partners can remain in the synchronized state during the maintenance window.
2. Perform manual failover of the database from the principal to the mirror. As the old mirror takes up the role of the new principal, the application can automatically reconnect to the new principal and continue processing, provided you have followed the best practices recommendations described in the white paper [Implementing Application Failover with Database Mirroring](http://www.microsoft.com/technet/prodtechnol/sql/bestpractice/implappfailover.mspx).
3. Once the planned maintenance is complete, you can fail back to the original principal.
4. Remove the database mirroring configuration.
5. Re-establish log shipping.

#### Log Shipping to a Third Destination from a Database Mirroring Pair

While database mirroring has many benefits, the one notable advantage of log shipping is its support of multiple secondaries. Those using log shipping with multiple secondaries need similar functionality with database mirroring in order to consider converting from log shipping to database mirroring.

Users want multiple copies of a database – usually one for local fault protection, and one or more at remote locations for disaster recovery. Database mirroring and log shipping can work together to provide a solution for this requirement. A mirrored pair (a principal and a mirror) can ship transaction log backups to a third server (or to many secondary servers). The principal server acts as the primary for the log shipping configuration. When a failover takes place between the mirrored pairs, the new principal (old mirror) seamlessly takes over the role of the primary server for the log shipping configuration.

A simplified architecture is shown in Figure 2. In this figure, Servers S1 and S2 are configured as database mirroring partners. One possible configuration may involve S1 and S2 in the same campus and configured with synchronous database mirroring to provide local fault protection; and server S3 at a remote site acting as the log shipping destination to provide disaster recovery solution. Another possible configuration may involve the three servers S1, S2 and S3 at three different sites.

****Figure 2: Log shipping to a third destination from a database mirroring pair****

##### **Set Up**

To set up log shipping to a third destination from a database mirroring pair

1. Setup database mirroring between S1 and S2. S1 is the principal and S2 is the mirror.
2. Before you set up log shipping, take a backup from S1 and restore it with NORECOVERY to S3.

**Note**: You can do this as part of step 3a; however, it is usually more convenient to do a backup and restore prior to setting up log shipping.

1. To set up log shipping from S1 (primary) to S3 (secondary):
2. Use the log shipping user interface in SQL Server Management Studio.
3. Before you click the final OK to establish log shipping, generate the script and save in a file. This script will be used in a later step. You can generate the script only before you click OK to establish log shipping. After you click OK, you cannot generate the script.
4. Click OK to establish log shipping.
5. Verify that log shipping from S1 to S3 is working. You can check the status of the log shipping jobs (backup, copy, and restore) in the SQL Agent job history. Make sure these jobs are running as scheduled without any errors.
6. If you are using asynchronous database mirroring between S1 and S2, change to synchronous database mirroring. Perform a manual failover of the database from S1 to S2. Now S2 is the principal and S1 is the mirror. Please note that failing over to the mirror database may impact the existing connections. The application can automatically reconnect to the new principal and continue processing, provided you have followed the best practices recommendations described in the white paper, [Implementing Application Failover with Database Mirroring](http://www.microsoft.com/technet/prodtechnol/sql/bestpractice/implappfailover.mspx).
7. Set up log shipping from S2 to S3, using S2 as the primary and S3 as the secondary.
8. Do not use the log shipping user interface in SQL Server Management Studio. The user interface allows only one primary for a given log shipping configuration. In this case, we are setting up two servers (although only one will be active at any given point in time) as the primary server. Instead, use the script generated in step 3b, but do not execute the entire script.
9. The generated script has two sections. One section is marked as “…Script to be run at Primary …”, and the other as “…Script to be run at Secondary …”. Execute only the section marked as “…Script to be run at Primary …” on S2. The section “…Script to be run at Secondary …” has already been executed on the secondary (S3) in step 3c, so there is no need to execute it again.

**Note**: Do not change anything in the script. Especially, make sure you use the same share location for backing up the transaction log as specified in the setup from S1 to S3.

1. Test to verify that log shipping from S2 to S3 is working.
2. Perform a manual failover of the database from S2 to S1. Again, S1 is the principal and S2 is the mirror.
3. If you changed to synchronous database mirroring in step 4, change back to asynchronous database mirroring.

This procedure sets up both S1 and S2 as the primary for log shipping. In the event of a database mirroring failover, the new principal will seamlessly take over the responsibility of the log shipping primary.

Log backup jobs are created on both S1 and S2. The log backup job on the principal (s1) will produce the log backup files on the share, which will be picked up by the copy job on the secondary.

The log backup job on the mirror (S2) will continue to execute, but will not produce anything. You will see the following message in the job status on the mirror:

A transactional log backup could not be generated because the database is in either NORECOVERY mode or STANDBY mode.

The message is self-explanatory and it is okay to get this message on the mirror.

To the secondary server (S3) it makes no difference which server (S1 or S2) is acting as the primary. The secondary server will always find the required log backup files on the same share location, and continue to apply them.

The configuration described here shows one secondary database. You can have several log shipping secondary databases.

##### **Potential Problem**

You should be aware of a potential problem that could arise when log shipping from a database mirroring pair to a third server. The log shipping secondary may experience an error condition if database mirroring failover takes place when “the mirroring hardened LSN" is less than “the log shipping backup LSN” (for details, see [Database Mirroring Sessions](http://msdn2.microsoft.com/en-us/library/ms189901.aspx)). LSN is the log sequence number (see [Introduction to Log Sequence Numbers](http://msdn2.microsoft.com/en-us/library/ms190411.aspx)). This can happen on rare occasions when something similar to the following happens:

* S1 has hardened LSNs 9, 10, 11.
* S2 has hardened LSNs 9, 10.
* The application has not yet received a confirmation that LSN 11 has been committed because S2 has not yet hardened.
* A failure on S1 causes failover to S2.
* The last log backup taken on S1 had LSN 11.
* The new log backup taken on S2 includes only LSN 10 (does not include LSN 11).
* The log backup taken on S2 now is out of sync with the secondary server (S3), because S3 has received the previous backup that included LSN 11.

You could see errors such as the following in the ERRORLOG of the secondary:

The backup data at the end of "D:\Backup\MyDB\_20071102214200.trn" is incorrectly formatted. Backup sets on the media might be damaged and unusable.

The log shipping secondary is now out of sync with the primary. You must rebuild the log shipping secondary, starting with a full backup from the primary. This will be resolved in a future release of SQL Server.

#### Switching Between the Database Mirroring Pair and the Log Shipping Pair

If you are log shipping to a third server from a database mirroring pair, and you need to switch which pair of servers are database mirroring partners and which is the log shipping secondary, you can do this fairly easily. One scenario where you might need to do this is when you do a rolling upgrade of all three servers. Figure 3 illustrates the before and after configuration.

****Figure 3: Switching between the database mirroring pair and the log shipping pair****

As illustrated in Figure 3, we start with a configuration where S1 and S2 are the database mirroring partners, and S3 is the log shipping secondary. In the target configuration, S1 and S3 are the database mirroring partners, and S2 is the log shipping secondary server.

To achieve the target configuration

1. Remove S2 as log shipping primary. Do not use the log shipping GUI from in SQL Server Management Studio. Instead, connect to the master database on the mirror server (S2), and run the following two stored procedures in the following order:
2. Execute **sp\_delete\_log\_shipping\_primary\_secondary** on S2.
3. Execute **sp\_delete\_log\_shipping\_primary\_database** on S2.
4. Stop the database mirroring session between S1 and S2.
5. Follow the steps in [Converting a Log Shipping Configuration to Database Mirroring](#_Converting_a_Log) to convert the log shipping configuration between S1 and S3 to a database mirroring configuration. S1 is now the principal and S3 is the mirror.
6. Follow the steps in [Log Shipping to a Third Destination from a Database Mirroring Pair](#_Log_Shipping_to) to set up log shipping to S2 from the database mirroring pair (S1 and S3). If you complete this conversion in a short period of time, and you have retained all the transaction log backups taken during this conversion, it is not necessary to perform a full restore on S2 while setting it up as a log shipping secondary.

#### Conclusion

Log shipping and database mirroring can work together to provide solutions for high availability and disaster recovery. You can convert an existing log shipping configuration to a database mirroring configuration. Although Microsoft only supports a single mirror database, you can set up log shipping from a database mirroring pair to additional servers, to provide multiple failover sites. Moreover, you can easily switch which pair of instances is using database mirroring and which instances are configured with log shipping.