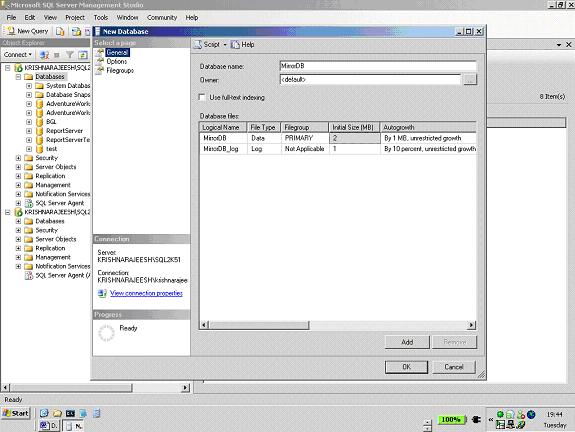
25 Nov

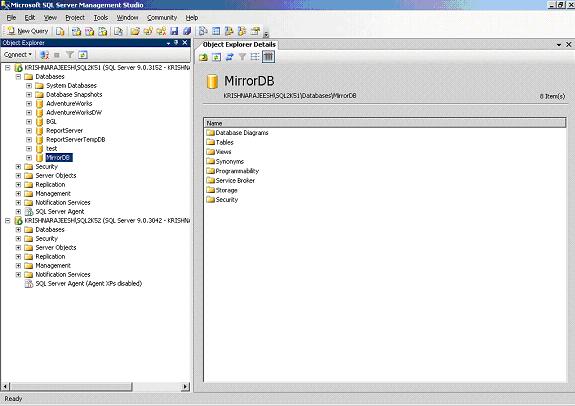
**Here is the detailed step by step implementation of Database Mirroring with screen shots. Note that I have shown here is a High Protection Mirroring scenario.**

**Here we are setting up a synchronous operating mode.**

**Step 1**

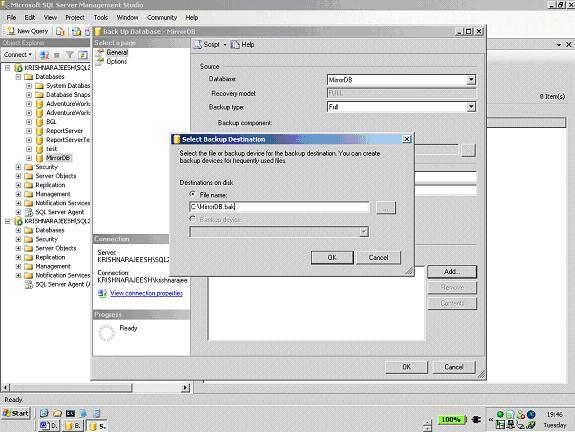
Create MirrorDB in the source database (called as Principal database), for which we are going to start mirroring.

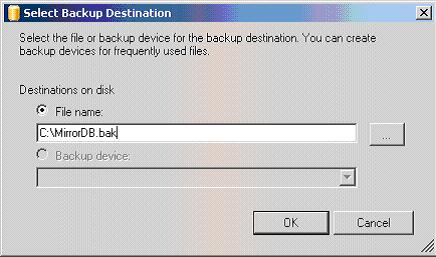


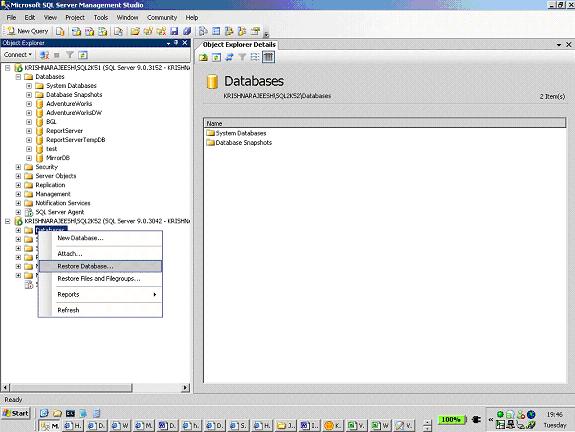


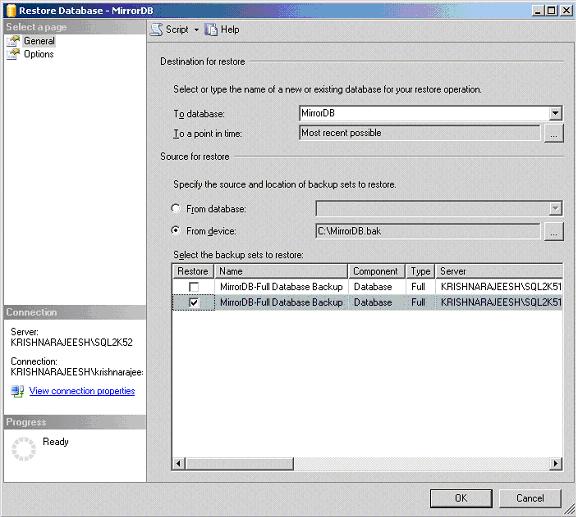
**Step 2**

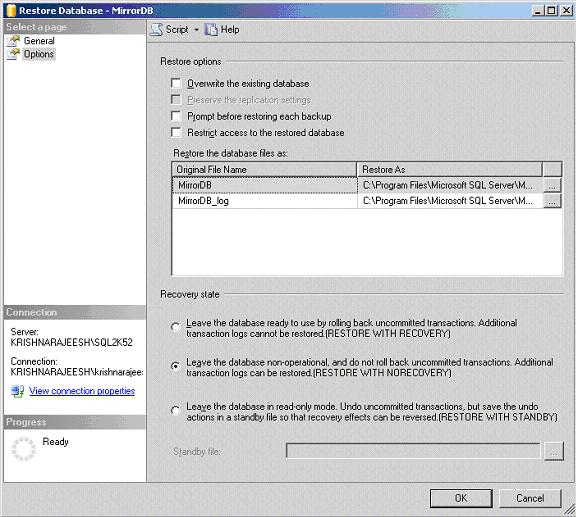
Create a backup of the MirrorDB in the source SQL Server (Principal database) & restore it in the destination server with the same name. Make sure to restore the database with NORECOVERY. The destination server MirrorDB is called as Mirror database.

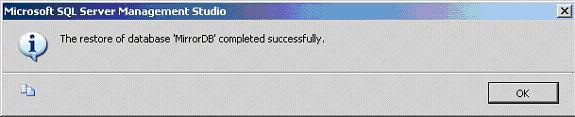










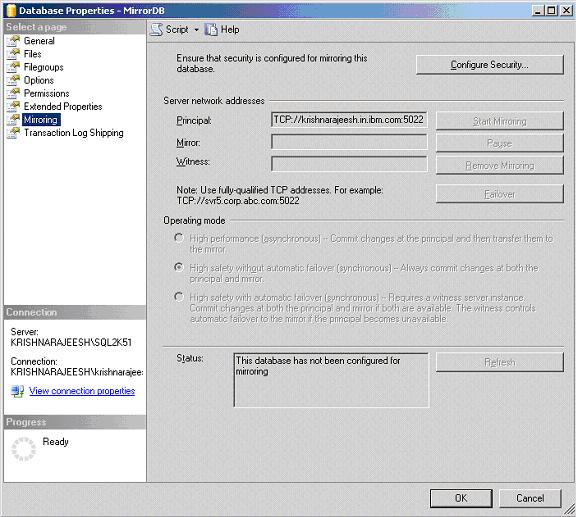


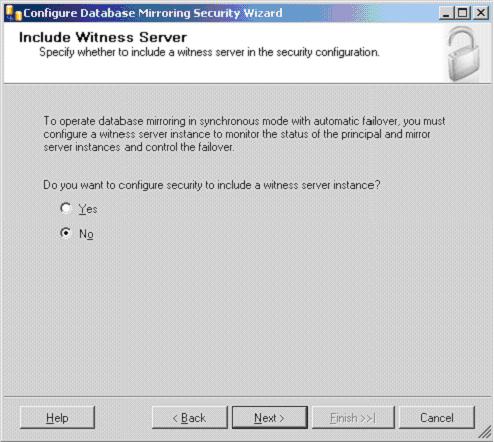
**Step 3**

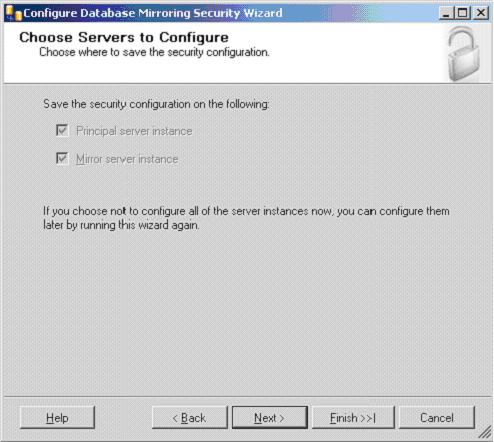
As shown in the below screen, right click on the database, select properties & select Mirroring. Select Configure Security to configure mirroring for the required database.

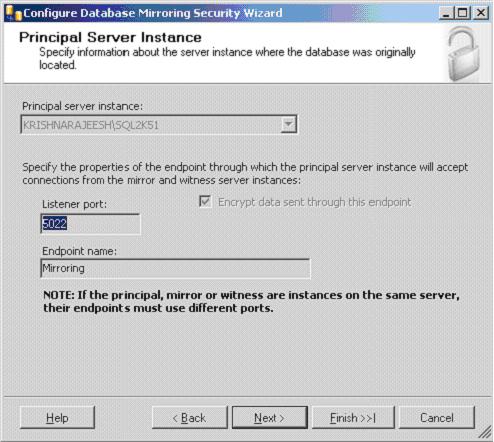
Follow the screen shots as given below.

 Note that here I do not have a Witness server, & hence select “NO” for include Witness server option.



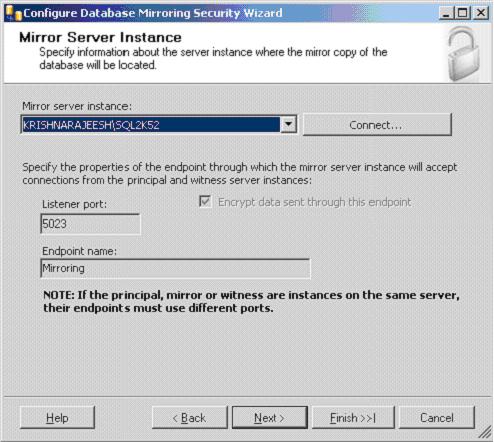






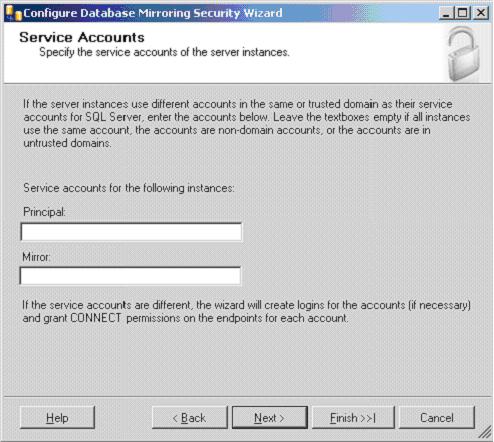
**Step 4**

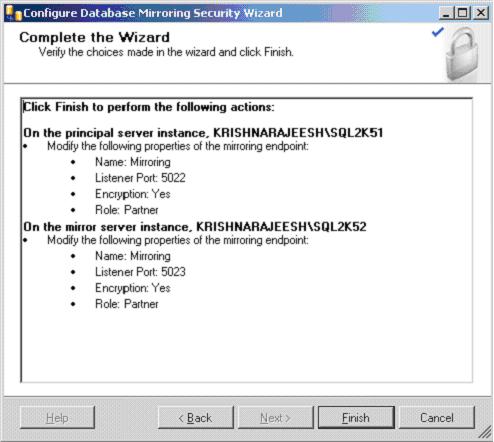
Select the Mirror Server Instance as shown below & click connect. Here we have to specify the destination, i.e. Mirror database server’s credentials to connect.

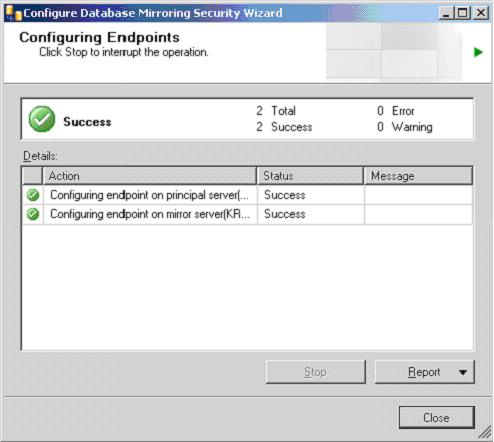


**Step 5**

Leave service accounts for both Principal & Mirror blank, follow the remaining screen shots.

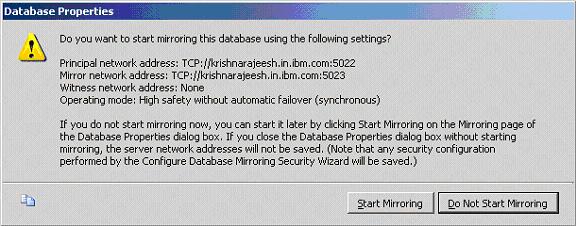






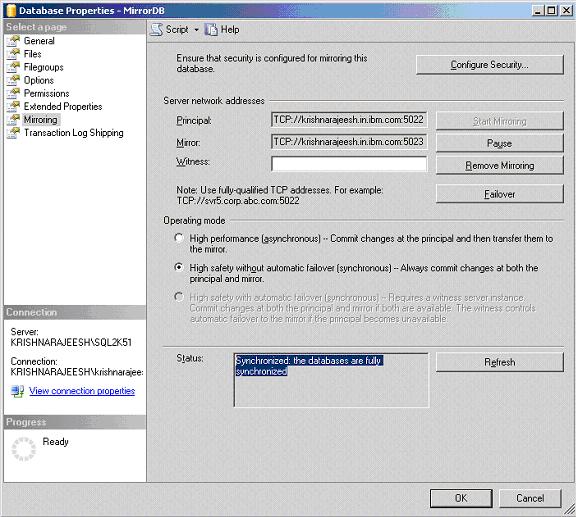
**Step 6**

Click on Start Mirroring to start the mirroring from Principal database to Mirror database.



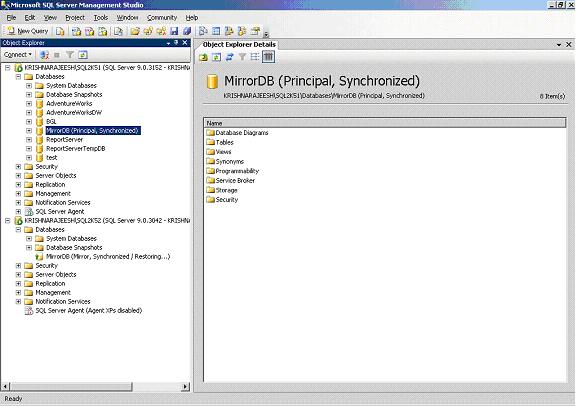
**Step 7**

As you can see in the below screen shot, the status is, the databases are fully Synchronized.



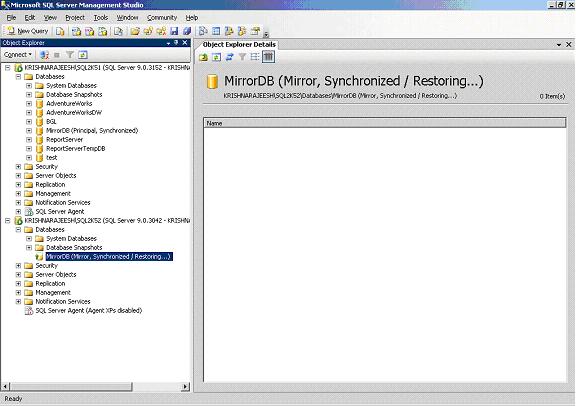
**Step 8**

The MirrorDB in the source server is now marked as **Principal, Synchronized**.



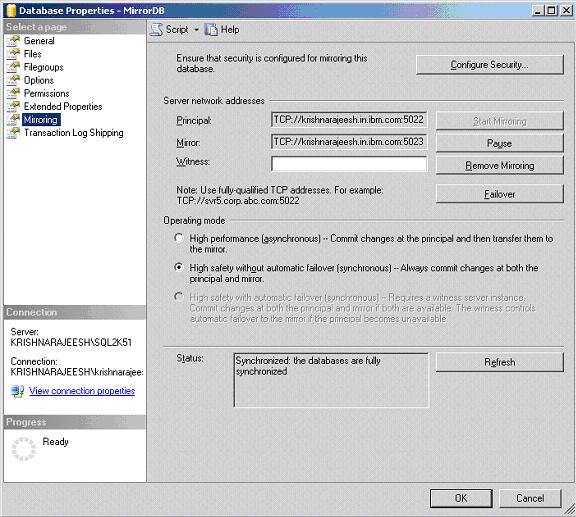
**Step 9**

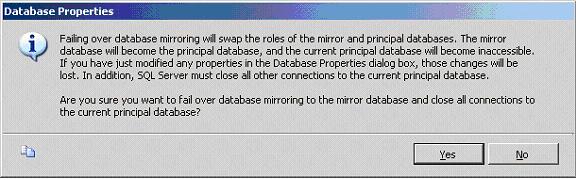
The MirrorDB in the destination server is now marked as **Mirror, Synchronized (Restoring)**

****

**Step 10**

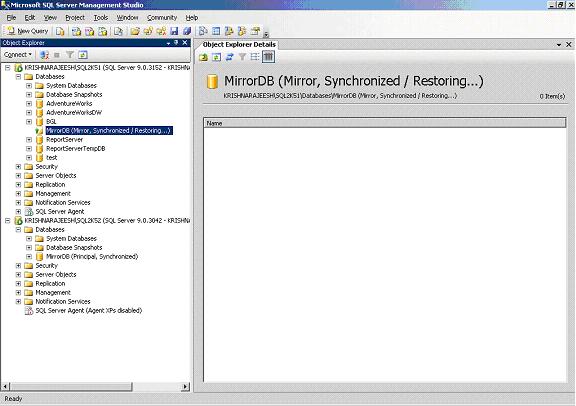
You can do manual failover as shown below, by selecting **Failover** option.

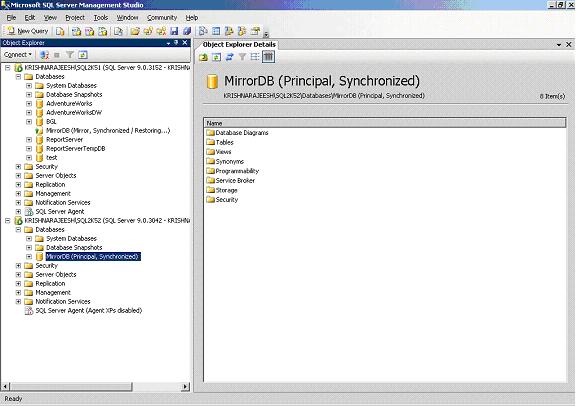




**Step 11**

The Principal database is now changed to Mirror & the Mirror is changed to Principal as shown in the below screen shots.





**Mirroring is a simple process & if failed we can easily reset it up, by restoring the latest backup of the source in the destination & reconfigure Mirroring.**

**DB Mirroring step by step & troubleshooting**

|  |
| --- |
| **Database mirroring** is a solution for increasing the availability of a SQL Server database. Mirroring is implemented on a per-database basis and works only with databases that use the full recovery model.  [**Benefits of Database Mirroring**](javascript:void(0);)**:**  Increases availability of a database: In the event of a disaster, in high-safety mode with automatic failover, failover quickly brings the standby copy of the database online (without data loss).  Increases data protection: Database mirroring provides complete or almost complete redundancy of the data, depending on whether the operating mode is high-safety or high-performance  Improves the availability of the production database during upgrades: To minimize downtime for a mirrored database, you can sequentially upgrade the instances of SQL Server that are hosting the failover partners. This will incur the downtime of only a single failover. This form of upgrade is known as a rolling upgrade.  **Requirements:**  **Principle** database recovery model should be full.  **Mirror** database recovery state should be no recovery.  **Endpoint**: Is nothing but a logical service point associated with a port no. It is a communication channel between servers in network.  **Types of Mirroring**:   We have 2 types of database mirroring  1) Asynchronous database mirroring.   2) Synchronous database mirroring.   Database Mirroring can be operated in 3 modes.   1) High performance.   2) High protection   3) High availability  **How to fail over Database Mirroring in SQL Server 2005 using T-SQL**  I've previously shown how to setup Database Mirroring in SQL Server 2005 with T-SQL, but how do you failover to the mirrored databases using T-SQL?  Here's how:  --Run on principal  USE master  GO  ALTER DATABASE dbName SET SAFETY FULL  GO  ALTER DATABASE dbName SET PARTNER FAILOVER  GO  --Run on new principal  USE master  GO  ALTER DATABASE dbName SET SAFETY OFF  GO  If you are using synchronous database mirroring, just ignore the "SET SAFETY" commands.  We use asynchronous database mirroring for performance reasons as our sites are 300 miles apart.  Even though we have a fast network in between the two sites, the latency is too high when we tried it in synchronous mode.     **Attached Related Documents  Below :** |

[Mirroring.doc](https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxzYWlmc3Fsc2VydmVycmVjaXBlc3xneDoxYmJiZGNmZTA5MGMxYmNi)

**Troubleshooting Log Shipping Issues**

30 *Wednesday* Dec 2009

Posted by [Amit Banerjee](http://troubleshootingsql.com/author/troubleshootingsql/) in [Troubleshooting SQL Issues](http://troubleshootingsql.com/category/troubleshooting-sql-issues/), [Wikis](http://troubleshootingsql.com/category/wikis/)

**≈** [**10 Comments**](http://troubleshootingsql.com/2009/12/30/troubleshooting-log-shipping-issues/#comments)

*Tags*

[*Log Shipping*](http://troubleshootingsql.com/tag/log-shipping/)

i

5 Votes

Quantcast

Log Shipping is a feature in SQL Server by which you can ship transaction log backups to a different server and restore the backups onto a standby database for disaster recovery or reporting purposes.

One of the major fundamental differences in SQL Server 2000 and SQL Server 2005 log shipping is that SQL Server 2005 uses linked servers to communicate between the primary and monitor and the secondary and monitor. The log shipping jobs are executed via linked server queries to update information about the **backup, copy and restore** jobs in case you have a remote monitor server.

So, if you find that your log shipping reports are defunct, then two additional things that you can do apart from the basic log shipping troubleshooting steps are:

1. Check if **remote connections** are enabled on the primary, secondary and monitor. For SQL Server 2000, check connectivity between the instances.

2. Check if a **linked server** using the same authentication settings as your log shipping setup can be setup between the primary and monitor or secondary or monitor depending on which part of your log shipping setup is broken. This is again applicable for SQL Server 2005.

**Basic Log Shipping Troubleshooting Steps**

1. Look into the SQL Server **ERRORLOGs**, **Application/System Event Logs** for any errors related to log shipping. Check the job history of the log shipping **Backup/Copy/Restore jobs** for any errors.

2. Check the **Backup, Copy & Restore job history details** for any errors.

3. If you are using SQL Server 2005, check what details are being displayed in the **Log Shipping Report** under **Standard Reports** in **Management Studio**.

4. If you want to take your troubleshooting to the next level, then you can even look into the log shipping meta data by querying the log shipping tables on the primary/secondary/monitor(if configured).

*Addendum: April 26th, 2011*

The log shipping configuration information can be found using the following methods:

1. Standard Reports – Transaction Log Shipping Status (right click Server Name in Management Studio-> Reports -> Standard Reports -> Transaction Log Shipping Status)  
2. Use the Stored Procedure which is called by the above report: **EXEC master..sp\_help\_log\_shipping\_monitor**

**Query to check log shipping job errors using the MSDB log shipping system tables**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | --List of Log Shipping jobs  SELECT \* from dbo.sysjobs WHERE category\_id = 6  SELECT \* FROM [msdb].[dbo].[sysjobhistory]  where [message] like '%Operating system error%'  order by [run\_date] , [run\_time]    SELECT \* FROM [msdb].[dbo].[log\_shipping\_monitor\_error\_detail]  where [message] like '%Operating system error%'    SELECT \* FROM [msdb].[dbo].[restorehistory] |

**Known issues with Log Shipping**

1. You might find that the last backed up/copied/restored files do not reflect correctly in the log shipping reports when you use a remote monitor server. In such a scenario, check if the following issue documented in the blog post below is applicable in your case:  
<http://blogs.msdn.com/b/sqlserverfaq/archive/2009/03/27/transaction-log-shipping-status-report-for-monitor-server-will-not-pull-up-information-if-alias-is-used-for-monitor-server.aspx>

The last copied and restored file will show up as null if the monitor instance is not on the same box as the secondary instance. The last backed up file will show up as null if the monitor instance is not on the same box as the primary instance if the **select @@servername** value is not used as the monitor server name while configuring the log shipping monitor.

2. If “**REMOTE ACCESS**” (sp\_configure will show if it is enabled or not) is not enabled or the **LOG SHIPPING LINKED SERVER** (to the monitor server) **is not working** for the primary and secondary servers, then last backup file/last copy file/last restored file information will not get populated if a remote monitor server instance is being used. The easiest way to identify this issue would be to capture a profiler trace (on primary instance when the backup job is running and on the secondary instance when the copy/restore job is running). The profiler trace will report errors if an update operation pertaining to the log shipping monitor tables fails provided all “Errors and Warnings” profiler events are captured.

3. Another issue that you could run into while using Log Shipping is **Orphaned Users** if you have Database Users on the Primary Database mapped to SQL Authenticated Logins. This happens because the SIDs of the SQL Authenticated Users on the Primary and Secondary instance would be different. I documented the workaround to this issue in the following blog post: <http://blogs.msdn.com/b/sqlserverfaq/archive/2009/04/13/orphaned-users-with-database-mirroring-and-log-shipping.aspx>

4. When you are scripting out an existing log shipping configuration, ensure that you have Cumulative Update Package 9 applied for SQL Server 2005 Service Pack 2 applied for Management Studio. If that is already done, then use one of the options mentioned in the **more information** section in the KB Article below:  
955693 FIX: In SQL Server 2005, the file information about the transaction log that was last copied and the file information about the transaction log that was last restored are missing  
<http://support.microsoft.com/default.aspx?scid=kb;EN-US;955693>

5. If you have configured LogShipping with STANDBY mode on SQL Server 2008 and the destination folder for the TLOGS uses a remote server (on which the **sqlservice/sqlagent** is not a Local Admin), then the restore job will fail everytime with following error :

2008-12-12 14:44:58.53 \*\*\* Error: During startup of warm standby database ‘testdb’ (database ID 7), its standby file (‘<UNC path of the TUF file>’) was inaccessible to the RESTORE statement. The operating system error was ’5(Access is denied.)’.

TUF = Transaction Undo File which is required for applying the next T-LOG backup. This issued is fixed in the cumulative update mentioned in the KB Article below:

FIX: Error message when you use log shipping in SQL Server 2008: “During startup of warm standby database ‘<Database Name>’ (database ID <N>), its standby file (‘<File Name>’) was inaccessible to the RESTORE statement”  
<http://support.microsoft.com/kb/962008>

6. Log shipping restore will fail if there is a snapshot or an active DBCC replica on the secondary database on which the restore is being done.  
<http://troubleshootingsql.com/2012/09/12/why-did-the-restore-fail-on-the-log-shipped-secondary-database/>

*Addition: September 12, 2012*

**Special cases**  
In case you need to speed up the transaction log restore for your log shipping secondary database in standby mode, then follow the steps mentioned in [this post](http://troubleshootingsql.com/2012/09/11/get-your-log-shipping-restore-to-run-faster/).

In case you need to move your secondary log shipped database files to a new physical location, then you can use the steps mentioned in [this post](http://troubleshootingsql.com/2012/09/13/moving-those-large-files-for-secondary-databases/).

**Disadvantages of Log Shipping**

Unfortunately, log shipping does have a few issues that the DBA should be aware of prior to relying on this method of failover:

1. Possible data loss when the primary server fails. If the primary server becomes completely unusable, transactions that occurred after the last transaction log backup that was copied to the standby server are lost. For example, suppose that server A fails at 5 a.m. and you cannot get to it at all. If the last backup copied to server B was taken at 4:45 a.m., all transactions that occurred between 4:45 a.m. and 5 a.m. are lost forever.
2. Some manual DBA work is required to bring the standby server online, as discussed in this article.
3. Log shipping setup cannot be scripted. This means that you cannot mimic the production environment for testing purposes without going through the wizard screens.
4. The Enterprise edition of SQL Server 2000 is required on primary and standby servers. If you run any other version/edition of SQL Server, you're out of luck. The Developer edition can be used to learn how to set up log shipping, but it cannot be used in a production environment. Note however, that log shipping is merely an automated way of copying transaction log backups, so a savvy DBA can easily set up jobs to accomplish the same functionality.
5. Difficult troubleshooting. Log shipping usually works very well, but if there are problems, they're difficult to troubleshoot—documentation is sparse and typically not helpful for solving a particular problem. Fortunately Microsoft's Knowledge Base articles have good information for troubleshooting log-shipping issues.
6. Each database that needs to be log-shipped must be set up through a separate maintenance plan.

http://stevenormrod.com/tag/troubleshooting/

As DBAs, we often obsess over backups. When was the backup made, how big is it, how quickly can I restore, can I restore it, etc. But sometimes, we need to know who made a backup.

Shouldn’t That Be Me? Well, of course, it should be you. But sometimes, it ain’t.

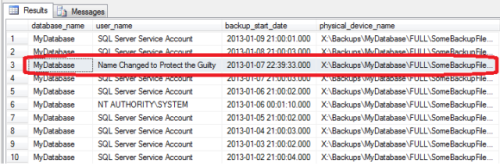
A few months back, I was asked to troubleshoot a SQL Server where they were getting disk full alerts. After a few quick checks, I was able to see it was the backup LUN that was filling up.

When I looked in the folder, I noticed there was a FULL backup file with an unusual timestamp. Most of the backup files were all created around the same time of day, but there was one with a completely different time. Next, I looked in the job history and this matched what I saw at the file system level.

OK, so now I knew what caused the space alerts, I still didn’t know who had caused them. Time to dig into the msdb database. msdb is a system database that contained all the job history for your SQL Server. A quick query revealed to me who it was that ran the ad-hoc backup.

[?](http://stevenormrod.com/tag/troubleshooting/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | -- look for who ran a manual backup  select top 30  s.database\_name, s.user\_name, s.backup\_start\_date,  m.physical\_device\_name  from msdb.dbo.backupset s  join msdb.dbo.backupmediafamily m  on s.media\_set\_id = m.media\_set\_id  where s.type = 'D'  and s.database\_name = 'YOUR\_DB\_NAME\_HERE'  order by s.backup\_start\_date desc  go |

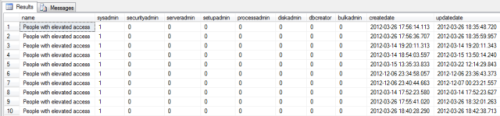
[](http://stevenormrod.com/wp-content/uploads/2013/05/who-made-the-backup.png)

Who Made the Backup

Next, I wanted to know how the user had access to create the backup. Let’s look at security. The first place I typically look is, who has admin access.

[?](http://stevenormrod.com/tag/troubleshooting/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | -- check server role membership  select l.name,  -- 'People with elevated access' as 'name',  l.sysadmin, l.securityadmin, l.serveradmin, l.setupadmin,  l.processadmin, l.diskadmin, l.dbcreator, l.bulkadmin,  l.createdate, l.updatedate  --, l.\*  from sys.syslogins l  where l.sysadmin = 1  or l.securityadmin = 1  or l.serveradmin = 1  or l.setupadmin = 1  or l.processadmin = 1  or l.diskadmin = 1  or l.dbcreator = 1  or l.bulkadmin = 1  order by l.name;  go |

[](http://stevenormrod.com/wp-content/uploads/2013/05/who-has-admin-access.png)

Who has Admin Access

And there’s my new best friend. Sometimes referred to as the Extra DBA, or People Who Can Get You Fired. In this case, there were several people who had been granted admin access. Time to start pruning some permissions.

The biggest hole on any system is extra people who have admin access. I see this all the time; perhaps a manager needs to run reporting queries, or developers need to maintain and troubleshoot an application.

When deadlines are looming and people are standing over your shoulder, the simplest fix often seems to just grant the person admin access. You’ll remember to remove it later, or when the app is deployed, right?

Posted in [SQL Server](http://stevenormrod.com/category/sql-server/) | Tagged [Backup](http://stevenormrod.com/tag/backup/), [DMVs](http://stevenormrod.com/tag/dmvs/), [Forensics](http://stevenormrod.com/tag/forensics/), [Security](http://stevenormrod.com/tag/security/), [Troubleshooting](http://stevenormrod.com/tag/troubleshooting/) | [Leave a reply](http://stevenormrod.com/2013/05/who-made-the-backup/#respond)

# [A Reboot Ate My Log Shipping](http://stevenormrod.com/2012/12/a-reboot-ate-my-log-shipping/)

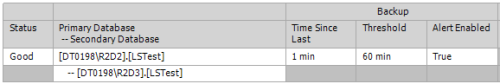
Posted on [**December 17, 2012**](http://stevenormrod.com/2012/12/a-reboot-ate-my-log-shipping/) by [**steven**](http://stevenormrod.com/author/steven/)

[Reply](http://stevenormrod.com/2012/12/a-reboot-ate-my-log-shipping/#respond)

I was recently contacted to troubleshoot some Log Shipping issues that arose after some SAN and server maintenance. The Server and SAN teams were working on the server; performing some maintenance on the MPIO drivers. After the work was finished and the server was rebooted, log shipping started to get out of sync on this server.

**First, Get a High-Level View, Then Dive Into the Details**

The best place to get a high-level view of Log Shipping is by using the built-in reports from SQL Server. In SSMS, right-click on the server node, select Reports, and then choose the Transaction Log Shipping Status report. Do this on both the primary and the secondary server in your log shipping configuration.

[](http://stevenormrod.com/wp-content/uploads/2012/12/ls-report-primary1.png)

Log Shipping Report – Primary Server

The Secondary Server’s report is a bit more informative in my opinion…

[Log Shipping Report - Secondary Server](http://stevenormrod.com/wp-content/uploads/2012/12/ls-report-secondary-alert1.png)

Log Shipping Report – Secondary Server

By viewing the report, I could see at a glance that the log backups were running fine, but the copy job was having trouble. Next I looked at the copy job history and saw the following error message:

* The network name cannot be found.

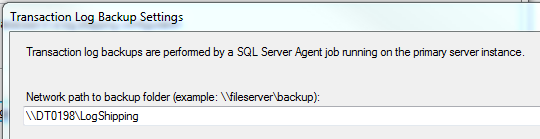
The name of the share can be found in two locations, the error log, or the Log Shipping Setup Properties. In the error log, look for this text in one of the substeps:

* Backup Source Directory
* Backup Destination Directory

[Log Shipping Errors](http://stevenormrod.com/wp-content/uploads/2012/12/ls-network-name-error1.png)

Log Shipping Errors

Or, in the Log Shipping Setup Properties look for this:

[](http://stevenormrod.com/wp-content/uploads/2012/12/ls-backup-share1.png)

Log Shipping Backup Share

From there, I logged onto the primary server and verified that the share was indeed missing. I also verified that the LOG backups were being written properly on the primary server, but missing from the secondary server.

To fix the situation, all that was necessary was to re-create the share on the primary server and grant read-only permissions to the secondary server’s service account.

After that, I manually ran the copy job to start the log backups flowing to the secondary server once again. Then, I ran the restore job and watched the secondary server get back in sync.

**But Don’t Rush In**

Often, when confronted with a Log Shipping sync problem, a person’s first reaction is to remove Log Shipping, and then reconfigure it. In this case, nothing would have been solved since the LOG share would have still been missing.

Instead, take the time to troubleshoot the problem and try to find the root cause. Log Shipping is a very simple and stable technology and not much will break it. I have never had to remove and reconfigure Log Shipping to get it to work again.

**Give Yourself Some Cushion**

Another key to success with Log Shipping is to keep the LOGs around long enough so that you don’t break the LSN chain in case of a problem. You want to make sure you keep enough LOGs on hand to cover a long weekend or holiday break in case your DBAs or other staff are not available to respond immediately.

[Log Shipping Retention](http://stevenormrod.com/wp-content/uploads/2012/12/delete-after-7-days.png)

Log Shipping Retention

My preference is to keep seven days worth of LOGs available. If you can’t get approval for that much space, walk it back a bit, but hold firm at three days worth. Think about the weekend/holiday factor. If management pushes back, remind them how long it took to initialize the secondary server with the FULL backup. Ask them which is worse, a little extra space, or a long delay while you get a 1TB FULL backup copied from one coast to another.

**But What About the Root Cause**

Oh yeah, so why did the share disappear to begin with? This server had been migrated from one SAN to another and ended up with two different vendor’s MPIO drivers present. So, the old one was slated to be removed. After the work was completed and the server rebooted, all of the shares went missing.

It turns out this is a known issue with the iSCSI Initiator if it is not configured correctly. Basically, the Server Service needs to have a dependency set on the iSCSI Initiator Service. Take a look at [kb870964](http://support.microsoft.com/kb/870964) for some more details on how to prevent this from happening to you.

Posted in [SQL Server](http://stevenormrod.com/category/sql-server/), [Storage](http://stevenormrod.com/category/storage/) | Tagged [iSCSI](http://stevenormrod.com/tag/iscsi/), [Log Shipping](http://stevenormrod.com/tag/log-shipping/), [Troubleshooting](http://stevenormrod.com/tag/troubleshooting/) | [Leave a reply](http://stevenormrod.com/2012/12/a-reboot-ate-my-log-shipping/#respond)

## [Log Shipping FAQ](http://www.brentozar.com/archive/2013/03/log-shipping-faq/)

I’ve gotten a lot of great questions about log shipping through these webcasts, so I’ve put together an FAQ.

* What editions of SQL Server is log shipping available in?
  + 2012 – Enterprise, Business Intelligence, Standard, and Web
  + 2008R2 – Datacenter, Enterprise, Standard, Web, and Workgroup
  + 2008 – Enterprise, Standard, Web, and Workgroup
  + 2005 – Enterprise, Standard, and Workgroup
* Does the secondary need to be licensed?
  + I am not the licensing police, and I am not Microsoft – check with your licensing representative to clarify your exact situation. Generally, you can have one warm standby server. However, the second someone starts using it for reporting, testing, or anything else, you need to license it like any other server.
* Log shipping is compatible with backup compression. What edition of SQL Server do I need to take advantage of compression?
  + 2012 – Enterprise, Business Intelligence, or Standard
  + 2008R2 – Datacenter, Enterprise, or Standard
  + 2008 – Enterprise
  + 2005 – Not available
* When log shipping is set up, Agent jobs are created to alert me if a backup, copy, or restore fails. How do I get notified?
  + You need to go into the Agent job, pull up Notifications, and choose your method – email an operator, or write to the event log, for example.
* Are my logins shipped from the primary to the secondary?
  + No, they are not. You’ll need to set up a separate method to sync the logins.
* Does this replace, or can it be combined with, our existing daily full and log backups?
  + TL; DR – no.
  + You’ll still want to take regular full and/or differential backups. Log shipping only takes one full backup – at the beginning – and that’s only if you specify that it does so. It can also be initialized from an existing full backup.
  + Taking two log backups in separate jobs will break the log chain, however. If you implement log shipping, it will replace your current transaction log backup job.
* What’s the difference between the secondary being in “Restoring” vs. “Standby”?
  + Restoring means the database is not accessible. Standby means it is read-only. You make this decision when you set up the log shipping.
  + If the database is in Standby mode, users can query it – except when a log backup is being restored. You need to decide if a restore job will disconnect users, or if the restore is delayed until after the users are disconnected