

Using SQL Backup Log Shipping to Maintain a Standby Server

Updated: February 2008

Introduction

This article provides an overview of log shipping, when to use it, and how Red Gate's SQL Backup simplifies the process. Some more detailed technical information about setting up log shipping using SQL Backup is then provided.

What is log shipping?

Log shipping provides a way of keeping in step databases that reside on different servers, or on different instances on the same server. It achieves this by automating the process of backing up data from a primary database on one server or instance and restoring the data to one or more secondary databases on another server or instance. You can then use the secondary server as a 'standby' for when your primary server is unavailable.



Transaction log backups are performed on the database on the primary server (for example the production database).



The backup files are copied to a shared folder that can be accessed by both servers.



The backup files are restored to the secondary database on the standby server.

When to use log shipping

You can use log shipping when you want to maintain a standby server but you do not require automatic failover.

Replication is another alternative for maintaining a standby server without automatic failover. For a detailed comparison of log shipping and replication to help you decide which is most appropriate for your requirements, see [Log Shipping vs Replication](#) by Paul Ibson on [SQLServerCentral.com](#).

Alternatively, you could use a failover clustering solution. This has the advantage of providing automatic failover. However, it is more difficult than log shipping to set up, requires more expensive licenses, and has other limitations such as on the locations of the servers.

Database mirroring provides another alternative, but it does not support multiple standby databases (unless it is used in conjunction with log shipping).

For more information, see [Configuring High Availability](#) in SQL Server Books Online.

Why use SQL Backup for log shipping?

Using SQL Backup makes log shipping much simpler compared with using native log shipping.

The SQL Backup graphical user interface provides a Log Shipping wizard to assist you with setting up the log shipping. The wizard guides you through the configuration.

Then, when you have reviewed a summary of the configuration, the Log Shipping wizard automatically makes a full database backup of the primary database and restores this backup on the standby database. This is to ensure that the standby database is consistent with the primary database and in the correct state for restoring the transaction log backups.

The wizard then sets up SQL Server Agent jobs that use SQL Backup to perform the backup and restore operations. A SQL Server Agent job is created on the primary server to periodically back up the transaction logs and copy the backup files to the shared folder; another job is created on the standby server to periodically retrieve the backup files from the shared folder and restore the transaction logs.

For full details of how to use the wizard, see the [SQL Backup help](#).

As well as simplifying the configuration process, SQL Backup is up to ten times faster than native backup and restore, and you can also compress the backups to save space, and encrypt the backups for added security.

Prerequisites

To perform log shipping using the SQL Backup Log Shipping wizard:

- You must have at least two SQL Server database engine servers or two database engine instances in your log shipping implementation.

Note that you cannot ship transaction logs from a SQL Server 2005 database to a SQL Server 2000 database, because of the differences in their structures (but you can ship from a SQL Server 2000 database to a SQL Server 2005 database if you use *No Recovery Mode* – see [Choosing the Log Shipping Mode](#) later in this article).

- SQL Backup Pro must be installed on both of the SQL Servers.
- The database must use the full or bulk-logged recovery model.
- You must have a shared folder to copy the transaction log backups to.

The SQL Server Agent service account of the primary server must have read/write access either to the shared folder or to the local NTFS folder. The SQL Server Agent account of the standby server must have read and delete access to the shared folder.

- The SQL Server Agent services must be running and configured with network credentials (such as a domain account) if you plan to use a network share as the shared folder.

You can configure log shipping with SQL Server Agent services stopped, but the process does not run until the agent is started.

Permissions

To use SQL Backup, the startup user for the SQL Backup Agent Service application must be a member of the SQL Server *sysadmin* fixed server role and have the following permissions:

- access to the shared folder
- the usual SQL Server backup permissions
- *execute* permissions on the SQL Backup extended stored procedure *sqlbackup* (for example, must be a member of the *public* database role in the master database)

- *execute* permissions on the following extended stored procedures:

<i>sqbdata</i>	<i>sqbstatus</i>
<i>sqbdir</i>	<i>sqbutility</i>
<i>sqbmemory</i>	<i>sqlbackup</i>

The user configuring the log shipping process must have *sysadmin* access to the participating servers.

Choosing the log shipping mode

When you use the SQL Backup Log Shipping wizard, you select the mode on Step 3 under **Recovery completion state**. The log shipping mode you select determines which operational requirement you want to use the technology for.

If you want to use the log shipping configuration only for the standby server, choose **Non-operational (No Recovery Mode)**; the standby database will not be available for users to query.

To allow users read-only access to the standby database, choose **Read only (Standby Mode)**; this means you can distribute queries across the standby server to reduce the primary server's workload (but you cannot modify the data in the standby database).

What happens during log shipping?

When log shipping is running, the SQL Server Agent backup job periodically makes a backup of the transaction logs on the primary server using SQL Backup. Each backup file is given a unique name.

The transaction log backup files are then copied to the shared folder.

The SQL Server Agent restore job periodically restores the completed transaction log backups on the shared folder to the standby server using SQL Backup. SQL Backup identifies backup groups or split backups, ensures all members of a backup group are present, arranges the files in sequence, and then restores each backup group sequentially.

The transaction log backups that have been restored are then moved from the shared folder to a location that you specify for processed backups when you configure the log shipping so that they are not picked up when the restore job is next run.

Customizing log shipping

The SQL Backup Log Shipping wizard provides an easy, step-by-step way to set up log shipping, but it may not provide all the options you need. If required, you can use the wizard to set up the jobs, and then manually modify the scripts that are contained in the backup and restore jobs to suit your needs.

For example, SQL Backup creates the standby database files in default locations. If you want to use an existing database as the standby database you may not want to use these default locations; you can edit the job so that it uses the standard Transact-SQL WITH MOVE argument to specify the locations for the files.

For more information about customizing the SQL Backup jobs, see [Log shipping using Red Gate SQL Backup](#) by Peter Yohz, and the [Toolkit](#) section of the [SQL Backup help](#).

Monitoring the log shipping

You can use the free Log Shipping Monitor tool for collating and summarizing SQL Server log shipping activity.

The Log Shipping Monitor provides customizable alerts for a variety of key log shipping variables, allowing you to take proactive steps to avoid any log shipping failures. For example,

you can set up an alert to notify you if the free space on a drive falls below a specified size, or if no transaction log backups have been created or restored on the standby server for a specified period of time.

In the event a log shipping process should fail, you can use the information provided by the Log Shipping Monitor to identify possible causes of the failure so that you can take the appropriate action.

You can download the Log Shipping Monitor from: http://labs.red-gate.com/index.php/Log_Shipping_Monitor

Failing over

If you need to switch from the primary database to a standby database, you must bring the standby database online manually; there is no automatic failover with the log shipping configuration.

The standby server will be inaccessible or read-only (depending upon whether you chose *No Recovery Mode* or *Standby Mode*). To make the standby database available for reading and writing, the log shipping needs to be broken and the database brought online WITH RECOVERY:

1. If possible, run the SQL Server Agent backup job on the primary server to perform a final log backup on the source database.

If the source database is damaged, a final log backup may fail. In this case, append NO TRUNCATE to the BACKUP LOG command in the SQL Server Agent backup job, and try running the job again.

2. If there are transaction log backup files remaining in the log shipping shared folder, use SQL Backup to restore the backups to the standby database, specifying the option WITH RECOVERY when you restore the final backup.

If there are no transaction log backup files remaining, recover the standby database using the native SQL Server RESTORE statement, for example:

```
RESTORE DATABASE [database_name] WITH RECOVERY
```

3. Reconfigure connecting resources to use the standby database.

If any SQL users were transferred via the transaction log backups, they will either not exist as logins on the standby server or they will use invalid security identifiers (SIDs) from the source server's login. To reconcile the SIDs:

1. Make a list of the users in the database's users container.
2. Open the security container and ensure that the users have corresponding logins on the server.
3. Reconcile all database users against the security identifier on the standby server using this query:

```
USE [MyDatabase]
exec sp_change_users_login 'Update_One', 'MyDBUser', 'MyDBLogin'
```

where *Update_One* links the specified user (*MyDBUser*) in the database (*MyDatabase*) to the existing SQL Server login (*MyDBLogin*). The user and the login are usually the same. For more information about *sp_change_users_login*, refer to *SQL Server Books Online*.

Note that bringing the database online breaks the log shipping. To re-establish log shipping, you must restore the standby database from a new full backup.

Summary

Log shipping provides a relatively easy way to keep a standby database in step with a production database, in case the production database becomes unavailable. It does not, however, provide automatic failover.

SQL Backup provides a wizard to guide you through the configuration, and to set up the SQL Server Agent jobs. Using SQL Backup and the Log Shipping Monitor, you can monitor the log shipping so that you can deal with any potential problems before they arise.

For more information about how to use the SQL Backup graphical user interface, see [Log Shipping](#) in the SQL Backup help. A [walkthrough](#) is also available on the Red Gate Web site.