

Configure Log Shipping (SQL Server)

Server 2014

- [SQL Server 2012](#)
- [SQL Server 2008 R2](#)
- [SQL Server 2008](#)
- [SQL Server 2005](#)

This topic describes how to configure log shipping in SQL Server 2014 by using SQL Server Management Studio or Transact-SQL.

Note

SQL Server 2008 Enterprise and later versions support backup compression. When creating a log shipping configuration, you can control the backup compression behavior of log backups. For more information, see [Backup Compression \(SQL Server\)](#).

Before You Begin

Prerequisites

- The primary database must use the full or bulk-logged recovery model; switching the database to simple recovery will cause log shipping to stop functioning.
- Before you configure log shipping, you must create a share to make the transaction log backups available to the secondary server. This is a share of the directory where the transaction log backups will be generated. For example, if you back up your transaction logs to the directory c:\data\logs\, you could create the \\primaryserver\logs share of that directory.

Security

Permissions

The log-shipping stored procedures require membership in the **sysadmin** fixed server role.

Using SQL Server Management Studio

To configure log shipping

1. Right click the database you want to use as your primary database in the log shipping configuration, and then click Properties.
2. Under Select a page, click Transaction Log Shipping.
3. Select the Enable this as a primary database in a log shipping configuration check box.
4. Under Transaction log backups, click Backup Settings.
5. In the Network path to the backup folder box, type the network path to the share you created for the transaction log backup folder.
6. If the backup folder is located on the primary server, type the local path to the backup folder in the If the backup folder is located on the primary server, type a local path to the folder box. (If the backup folder is not on the primary server, you can leave this box empty.)

Important

If the SQL Server service account on your primary server runs under the local system account, you must create your backup folder on the primary server and specify a local path to that folder.

7. Configure the Delete files older than and Alert if no backup occurs within parameters.
8. Note the backup schedule listed in the Schedule box under Backup job. If you want to customize the schedule for your installation, then click Schedule and adjust the SQL Server Agent schedule as needed.
9. SQL Server 2014 supports [backup compression](#). When creating a log shipping configuration, you can control the backup compression behavior of log backups by choosing one of the following options: Use the default server setting, Compress backup, or Do not compress backup. For more information, see [Log Shipping Transaction Log Backup Settings](#).
10. Click OK.
11. Under Secondary server instances and databases, click Add.
12. Click Connect and connect to the instance of SQL Server that you want to use as your secondary server.
13. In the Secondary Database box, choose a database from the list or type the name of the database you want to create.
14. On the Initialize Secondary database tab, choose the option that you want to use to initialize the secondary database.

Note

If you choose to have Management Studio initialize the secondary database from a database backup, the data and log files of the secondary database are placed in the same location as the data and log files of the **master** database. This location is likely to be different than the location of the data and log files of the primary database.

15. On the Copy Files tab, in the Destination folder for copied files box, type the path of the folder into which the transaction logs backups should be copied. This folder is often located on the secondary server.
16. Note the copy schedule listed in the Schedule box under Copy job. If you want to customize the schedule for your installation, click Schedule and then adjust the SQL Server Agent schedule as needed. This schedule should approximate the backup schedule.
17. On the Restore tab, under Database state when restoring backups, choose the No recovery mode or Standby mode option.
18. If you chose the Standby mode option, choose if you want to disconnect users from the secondary database while the restore operation is underway.
19. If you want to delay the restore process on the secondary server, choose a delay time under Delay restoring backups at least.
20. Choose an alert threshold under Alert if no restore occurs within.
21. Note the restore schedule listed in the Schedule box under Restore job. If you want to customize the schedule for your installation, click Schedule and then adjust the SQL Server Agent schedule as needed. This schedule should approximate the backup schedule.
22. Click OK.
23. Under Monitor server instance, select the Use a monitor server instance check box, and then click Settings.

Important

To monitor this log shipping configuration, you must add the monitor server now. To add the monitor server later, you would need to remove this log shipping configuration and then replace it with a new configuration that includes a monitor server.

24. Click Connect and connect to the instance of SQL Server that you want to use as your monitor server.
25. Under Monitor connections, choose the connection method to be used by the backup, copy, and restore jobs to connect to the monitor server.
26. Under History retention, choose the length of time you want to retain a record of your log shipping history.
27. Click OK.
28. On the Database Properties dialog box, click OK to begin the configuration process.

Using Transact-SQL

To configure log shipping

1. Initialize the secondary database by restoring a full backup of the primary database on the secondary server.
2. On the primary server, execute [sp_add_log_shipping_primary_database](#) to add a primary database. The stored procedure returns the backup job ID and primary ID.
3. On the primary server, execute [sp_add_jobschedule](#) to add a schedule for the backup job.
4. On the monitor server, execute [sp_add_log_shipping_alert_job](#) to add the alert job.
5. On the primary server, enable the backup job.
6. On the secondary server, execute [sp_add_log_shipping_secondary_primary](#) supplying the details of the primary server and database. This stored procedure returns the secondary ID and the copy and restore job IDs.
7. On the secondary server, execute [sp_add_jobschedule](#) to set the schedule for the copy and restore jobs.
8. On the secondary server, execute [sp_add_log_shipping_secondary_database](#) to add a secondary database.
9. On the primary server, execute [sp_add_log_shipping_primary_secondary](#) to add the required information about the new secondary database to the primary server.
10. On the secondary server, enable the copy and restore jobs. For more information, see [Disable or Enable a Job](#).

Related Tasks

- [Upgrade Log Shipping to SQL Server 2014 \(Transact-SQL\)](#)
- [Add a Secondary Database to a Log Shipping Configuration \(SQL Server\)](#)
- [Remove a Secondary Database from a Log Shipping Configuration \(SQL Server\)](#)
- [Remove Log Shipping \(SQL Server\)](#)
- [View the Log Shipping Report \(SQL Server Management Studio\)](#)
- [Monitor Log Shipping \(Transact-SQL\)](#)
- [Fail Over to a Log Shipping Secondary \(SQL Server\)](#)

The correct order to configure log shipping must be:

1. On the primary server, execute [sp_add_log_shipping_primary_database](#) to add a primary database. The stored procedure returns the backup job ID and primary ID.
 2. On the primary server, execute [sp_add_jobschedule](#) to add a schedule for the backup job.
 3. On the monitor server, execute [sp_add_log_shipping_alert_job](#) to add the alert job.
 4. On the primary server, enable the backup job.
 5. Initialize the secondary database by restoring a full backup of the primary database on the secondary server, with NORECOVERY or STANDBY options.
 6. On the secondary server, execute [sp_add_log_shipping_secondary_primary](#) supplying the details of the primary server and database. This stored procedure returns the secondary ID and the copy and restore job IDs.
 7. On the secondary server, execute [sp_add_jobschedule](#) to set the schedule for the copy and restore jobs.
 8. On the secondary server, execute [sp_add_log_shipping_secondary_database](#) to add a secondary database.
 9. On the primary server, execute [sp_add_log_shipping_primary_secondary](#) to add the required information about the new secondary database to the primary server.
 10. On the secondary server, enable the copy and restore jobs. For more information, see [Disable or Enable a Job](#).
-