Managing SQL Transaction Log File Growth

https://campus.barracuda.com/doc/98217075/

In some cases, the log file for your Service Center database can grow to an unmanageable size, and you would like to recover some disk space. The file is named *SCData_default.ldf*.

Microsoft SQL Server uses transaction logs in order to increase I/O performance. Transactions are written to the log file faster than they can be written to the database, and are then copied into the database by SQL Server in the background. Service Center supports two SQL Server recovery models: Simple Recovery and Full Recovery.

Simple Recovery

Simple Recovery logs uncommitted transactions, which are those that have not yet been written to the database. Once a transaction has been written to the database, or committed, the log entry is removed. This does not reduce the disk space consumed by the log file, however, as SQL Server will reclaim and use the space within the log file but will not release space back to Windows.

The size of the log file will grow to its maximum size based on your monitoring configuration and the total number of devices being managed and remain fairly consistent. Use the SQL Server to see the available space within a log file.

When you add new Sites, devices, or monitors you can expect to see a relative amount of growth in the size of the log. Unhealthy devices that generate frequent Windows events, SNMP traps, or syslogs can also cause the log file to consume more space. The space used by the log will always increase in size and will never shrink on its own. Running a backup will not shrink the log file, so when using Simple Recovery a shrink should be performed should the log file increase in size significantly month to month.

Using Simple Recovery mode you will usually see much less disk space taken up by the log files.

Full Recovery

Full Recovery logs store committed and uncommitted transactions. Once a transaction has been written to the database, or committed, the log entry is not removed. This causes the log file to grow until the next successful backup occurs. When the backup completes, the space occupied by committed transactions is reclaimed within the log file. Size will appear unchanged unless a shrink operation is performed.

When you add new Sites, devices or monitors you can expect to see a relative amount of growth in the size of the log very rapidly, especially when unhealthy devices generate extra monitoring data.

Running a backup will not shrink the database, nor will a manual shrink significantly reduce the overall size of the log. The only reduction in the size of the log file will come from successful backups combined with a shrink operation. This means that in order to contain the log file growth, you must schedule more frequent backups, at least once a day.

A full backup is used in conjunction with a replay of the log file, to recover the database without any loss of data, providing the log file was not lost.

Switching From Full To Simple Recovery Models

While switching from full to simple recovery will allow you to reclaim disk space, it will also break the backup log chain. To deal with this, we strongly recommend that you backup the log immediately prior to making the change, which will then allow you to restore the database up to that point. After making the switch you will need to take periodic backups to protect your data and to truncate the inactive portion of the transaction log.

1. Launch **SQL Management Studio** and connect to the Service Center database server.
2. Expand **Databases** and right-click **SCData_Default**. Select **Properties**.
3. Select the **Options** page and select **Simple** as the **Recovery Model**, then click **OK**.
4. Right-click **SCData_default** and select **Tasks** > **Shrink** - **Files**.
5. From the **General** page, select **Log** as the **File type**.
   The current allocated space and the available free space (the amount you will recover) are shown.
6. Choose **Release unused space** as the **Shrink action**. Click **OK**.
7. Right-click **SCData_default** and select **Tasks** > **Backup**.
8. From the **General** page choose your **Backup type** (**Full** or **Differential**) and Destination and click **OK**. The backup process will also reduce the size of the log file.

Create a Maintenance Plan in SQL

Performing a backup of SCData_default as part of regular maintenance will also help to keep the size of the transaction log down to a manageable amount. Depending on your environmental needs, you may wish to also include a shrink, but remember when you do so you’re are trading the disk footprint for SQL performance speeds, as SQL must create white space to which records are written, and shrinking eliminates this white pace.

The maintenance plan below is an example only. Depending on your situation you may need to make more or less frequent backups and add or remove other tasks.

1. Launch SQL **Management Studio** and connect to the Service Center database server.
2. Expand Management and right-click **Maintenance Plans**. Select **Maintenance Plan Wizard**.
3. Click **Next**.
4. Enter a meaningful **Name** for the plan and select **Separate schedules for each task**. Click **Next**.
5. Select the tasks you wish to schedule and click **Next**.
6. Because you selected separate schedules for each task, click **Next** when asked for the order for the tasks to execute.
7. Select the **SCData_default database** and click **OK**.
    - To configure a **Shrink Database** task, choose 10% as the amount of free space to remain after the shrink, and select **Return freed space to operating system**.
    - To configure a **Backup Database (Full)** task, choose a location (Disk or Tape) and to **Overwrite existing backup files**.

      > The recommended schedule is full backups once a week, with differential backups every other day.
    - To configure a **Backup Database (Differential)** task, choose a location (**Disk** or **Tape**) and to **Overwrite existing backup files**.

      > Do not schedule a differential on the same day you are performing a full backup.
    - To configure a **Backup Database (Transaction Log)** task, choose a location (**Disk** or **Tape**) and to **Overwrite existing backup files**.

      > **Tips**
      >
      > - If you have set the recovery model to Simple, you do not need to perform this task. It is only required when leaving the recovery model at Full.
      > - Backing up the transaction log once a week, at a time when no other maintenance operations are running, is recommended.
8. Click **Change** to open the **Job Schedule Properties**. Set a schedule that suits your needs. Click **OK** and then **Next**.
9. Choose to **Write a report to a test file** (suitable for monitoring with a Barracuda RMM Custom Log Monitor) or to **E-mail the report** if you wish.
10. Click **Next**, and then click **Finish**.

Additional Information

Recovery Models for System Databases

https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/back-up-and-restore-of-system-databases-sql-server?view=sql-server-2017

Consult SQL Server Documentation

https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation?toc=..%2Ftoc.json

&view=sql-server-2017:

- Managing a Full Transaction Log
- Shrinking the Transaction Log
- Shrinking a Database
- Using Transaction Log Backups