[September 20, 2010](http://blog.sqlauthority.com/2010/09/20/sql-server-how-to-stop-growing-log-file-too-big/" \o "7:00 am) by [Pinal Dave](http://blog.sqlauthority.com/author/pinaldave/)

**SQL SERVER – How to Stop Growing Log File Too Big**

I was recently engaged in [**Performance Tuning**](http://blog.sqlauthority.com/sql-server-performance-tuning/) Engagement in Singapore. The organization had a huge database and had more than a million transactions every hour. During the assignment, I noticed that they were truncating the transactions log. This really alarmed me so I informed them this should not be continued anymore because there’s really no need of truncating or shortening the database log. The reason why they were truncating the database log was that it was growing too big and they wanted to manage its large size. I provided two different solutions for them. Now let’s venture more on these solutions.

If you are jumping over this post to leave a comment, please read first the two options as follows:

**1) Convert the Recovery Model to Simple Recovery**

If you are truncating the transaction logs, this means you are breaking the T-Log LSN (Log Sequence Numbers). This follows that if disaster comes, you would not be able to restore your T-Logs and there would be no option for you to do point in time recovery. If you are fine with this situation and there is nothing to worry, I suggest that you change your recovery model to Simple Recovery Model. This way, you will not have extra ordinary growth of your log file.

**2) Start Taking Transaction Log Backup**

If your business does not support loss of data or requires having point in time recovery, you cannot afford anything less than Full Recovery Model. In Full Recovery Model, your transaction log will grow until you take a backup of it. You need to take the T-Log Backup at a regular interval. This way, your log would not grow beyond some limits. If you are taking an hourly T-Log backup, your T-Log would grow until one hour but after this the T-Log backup would truncate all the ‘committed’ transactions once you take it. Doing this would lead the size of the T-Log not to go down much, but it would rather be marked as **empty** for the next hour’s T-Log to populate.

With this method, you can restore your database at Point of Time if a disaster ever happens at your server.

Let us run an example to demonstrate this. In this case, I have done the following steps:

1. Create Sample Database in FULL RECOVERY Model
2. Take Full Backup (full backup is must for taking subsequent backup)
3. Repeat Following Operation
   1. Take Log Backup
   2. Insert Some rows
   3. Check the size of Log File
4. Clean Up

After a short while, you will notice that the Log file (ldf) will stop increasing but the size of the backup will increase.

If you have an issue with your log file growth, I suggest that you follow either of the above solutions instead of truncating it.

https://support.microsoft.com/en-us/kb/317375