**Previous videos**

• What is a transaction log file --file to record all changes to the database  
• Why backup transaction log file -- to clear the transaction log and control size  
• Inside the transaction log file -- Use DBCC LogInfo  
• Out of control transaction log file -- primary reason, full recovery mode and no transactional backups

**What is the auto growth feature?**

An auto-growth event is a part of SQL Server that expands the size of a database file when it runs out of space. If there is a transaction (such as many inserts) that requires more log space than is available, the transaction log file of that database will need to adjust to the new space needed by increasing the log file size.

**What happens when auto growth is expanding?**

This can cause a performance issue, as this is a blocking operation. The transaction that that initiated the log growth will be held until more space is allocated to the log file, determined by the auto growth setting

Physical fragmented on the disk occurs as the pages required are not necessarily next to each other. The more auto-growth events you have the more physical fragmentation you will have the files

**Avoid auto growth by pro actively configuring the auto growth**

Pre-size the data and log files

Manually manage the growth of data and log files

Auto growth should be used for safety reasons only

**Don’t rely on auto growth**

Maintain a level of at least 25 percent available space across disks to allow for growth and peak usage patterns

Set the auto-grow settings to grow based on megabytes instead of a percentage, so as to have the auto growth consistent

--Drop database auto

--Drop database auto2

--Create database with default setting based on the Model database configuration

Use master

go

CREATE DATABASE [auto]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'auto',

FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\auto.mdf' ,

SIZE = 3072KB , --<< initial size of data file 3mb

FILEGROWTH = 1024KB ) --<< growth by 1mg

LOG ON

( NAME = N'auto\_log',

FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\auto\_log.ldf' ,

SIZE = 1024KB , --<< initial size of log file 1mb

FILEGROWTH = 10%) --<< growth by 10%

GO

DBCC LogInfo;

--Create database with set LOG FILE setting

CREATE DATABASE [auto2]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'auto2',

FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\auto2.mdf' ,

SIZE = 1024000KB , --<< initial size of data file 1000mb

FILEGROWTH = 102400KB )

LOG ON

( NAME = N'auto2\_log',

FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\auto2\_log.ldf' ,

SIZE = 102400KB ,

FILEGROWTH = 102400KB ) --<< growth by 100mb (PRE SIZED SO THAT THE AUTO GROWTH DOES NOT ACTIVATE)

GO

DBCC LogInfo

-- examine the database files

sp\_helpdb auto

sp\_helpdb auto2

dbcc sqlperf (logspace)

--move data from adventureworks2012 to auto and auto2 dtabase via import/export wizard OR

Select \* Into LogGrowthTable from adventureworks2012.sales.SalesOrderDetai3l

------------------------------------------------------------------------

--RESULTS:

--As the insert is being recored in the transaction log that was 1mb in size, the initial size (1mb) of the tlog recording can't keep up with the

--activity, and as such, needs to expand by 10% each time there is modifications to record.

-- query to find auto growth setting for all or specified database (or use the SQL reports)

USE [master]

GO

BEGIN TRY

IF (SELECT CONVERT(INT,value\_in\_use) FROM sys.configurations WHERE NAME = 'default trace enabled') = 1

BEGIN

DECLARE @curr\_tracefilename VARCHAR(500);

DECLARE @base\_tracefilename VARCHAR(500);

DECLARE @indx INT;

SELECT @curr\_tracefilename = path FROM sys.traces WHERE is\_default = 1;

SET @curr\_tracefilename = REVERSE(@curr\_tracefilename);

SELECT @indx = PATINDEX('%\%', @curr\_tracefilename) ;

SET @curr\_tracefilename = REVERSE(@curr\_tracefilename) ;

SET @base\_tracefilename = LEFT( @curr\_tracefilename,LEN(@curr\_tracefilename) - @indx) + '\log.trc';

SELECT

--(DENSE\_RANK() OVER (ORDER BY StartTime DESC))%2 AS l1,

ServerName AS [SQL\_Instance],

--CONVERT(INT, EventClass) AS EventClass,

DatabaseName AS [Database\_Name],

Filename AS [Logical\_File\_Name],

(Duration/1000) AS [Duration\_MS],

CONVERT(VARCHAR(50),StartTime, 100) AS [Start\_Time],

--EndTime,

CAST((IntegerData\*8.0/1024) AS DECIMAL(19,2)) AS [Change\_In\_Size\_MB]

FROM ::fn\_trace\_gettable(@base\_tracefilename, default)

WHERE

EventClass >= 92

AND EventClass <= 95

--AND ServerName = @@SERVERNAME

--AND DatabaseName = 'myDBName'

AND DatabaseName IN ('auto','auto2')

ORDER BY DatabaseName, StartTime DESC;

END

ELSE

SELECT -1 AS l1,

0 AS EventClass,

0 DatabaseName,

0 AS Filename,

0 AS Duration,

0 AS StartTime,

0 AS EndTime,

0 AS ChangeInSize

END TRY

BEGIN CATCH

SELECT -100 AS l1,

ERROR\_NUMBER() AS EventClass,

ERROR\_SEVERITY() DatabaseName,

ERROR\_STATE() AS Filename,

ERROR\_MESSAGE() AS Duration,

1 AS StartTime,

1 AS EndTime,

1 AS ChangeInSize

END CATCH

--DBCC LogInfo;