

# TempDB how its works?



Dubi Lebel  
Dubi Or Not To Be  
Dubi.Lebel@gmail.com

# How this presentation start?

---

- Sizing Application
- Application databases
- **TempDB – size & IOPS?**

# November 2009

**Nothing is not more  
permanent than the  
temporary (A. Kishon)**

Wonder TempDB

Dubi Lebel

Dad, chef, gardener and  
DBA  
Dubi.lebel@gmail.com

# What I knew

---

- Only one TempDB per instance.
- Used often implicitly by all connections.
- Re create when server restart.
- Operations within TempDB are **minimally logged**.
- Data in TempDB does not persist after SQL Server shuts down.
- Only one file group in TempDB is allowed for data and one file group for logs.
- By default auto grow is enabled.
- When the server restarts, the TempDB file size is reset to the configured value (the default is 8MB).

## Re create Tempdb

---

- Creating new database based on Model DB and its properties.
- Lock tempdb and model, no one can connect to server?
- Create MDF(4MB as the model).
- Copy extents from model MDF.
- Update system data.
- Create LDF (5 MB as model).
- If exists other files create other NDF.
- Extend MDF and LDF to the last configure size.

Load Log
 Export
 Refresh
 Filter ...
 Search ...
 Stop
 Help

Log file summary: No filter applied

Date ▼	Source	Message
3/29/2016 16:59:27 PM	spid24s	1 transactions rolled forward in database 'DBCC_DEMO' (6:0). This is an informational message only. No user action is required.
3/29/2016 16:59:09 PM	spid29s	Service Broker manager has started.
3/29/2016 16:59:09 PM	spid29s	The Database Mirroring endpoint is in disabled or stopped state.
3/29/2016 16:59:09 PM	spid29s	The Service Broker endpoint is in disabled or stopped state.
3/29/2016 16:59:07 PM	spid10s	Starting up database 'tempdb'. —
3/29/2016 16:59:04 PM	spid10s	Clearing tempdb database. —
3/29/2016 16:59:04 PM	spid7s	0 transactions rolled back in database 'AdventureWorks2014' (5:0). This is an informational message only. No user action is required.
3/29/2016 16:59:03 PM	spid23s	6 transactions rolled forward in database 'AdventureWorks2014' (5:0). This is an informational message only. No user action is required.
3/29/2016 16:59:03 PM	spid10s	Starting up database 'model'. —
3/29/2016 16:59:02 PM	spid7s	Recovery completed for database 'Vulcan' (database ID 8) in 1 second(s) (analysis 103 ms, redo 23 ms, undo 809 ms.) This is an informational message only. No user action is required.
3/29/2016 16:59:02 PM	spid7s	Recovery is writing a checkpoint in database 'Vulcan' (8). This is an informational message only. No user action is required.
3/29/2016 16:59:02 PM	spid7s	0 transactions rolled back in database 'Vulcan' (8:0). This is an informational message only. No user action is required.
3/29/2016 16:59:01 PM	spid26s	272 transactions rolled forward in database 'Vulcan' (8:0). This is an informational message only. No user action is required.
3/29/2016 16:59:01 PM	spid7s	0 transactions rolled back in database 'msdb' (4:0). This is an informational message only. No user action is required.
3/29/2016 16:59:01 PM	spid22s	1 transactions rolled forward in database 'msdb' (4:0). This is an informational message only. No user action is required.
3/29/2016 16:59:01 PM	spid7s	0 transactions rolled back in database 'SSISDB' (9:0). This is an informational message only. No user action is required.
3/29/2016 16:59:01 PM	spid7s	0 transactions rolled back in database 'PartitoinDemo' (7:0). This is an informational message only. No user action is required.
3/29/2016 16:59:00 PM	spid27s	2 transactions rolled forward in database 'SSISDB' (9:0). This is an informational message only. No user action is required.
3/29/2016 16:59:00 PM	spid25s	1 transactions rolled forward in database 'PartitoinDemo' (7:0). This is an informational message only. No user action is required.
3/29/2016 16:59:00 PM	spid10s	The resource database build version is 12.00.4100. This is an informational message only. No user action is required.
3/29/2016 16:58:59 PM	spid28s	Starting up database 'UG_Demo'.
3/29/2016 16:58:59 PM	spid10s	Starting up database 'mssqlsystemresource'.
3/29/2016 16:58:59 PM	spid24s	Starting up database 'DBCC_DEMO'.
3/29/2016 16:58:59 PM	spid27s	Starting up database 'SSISDB'.
3/29/2016 16:58:59 PM	spid26s	Starting up database 'Vulcan'.

# Create Tempdb

---

- Initialization data file vs. log file
- What will happened if server can not create TempDB?
- Start server with -F
- When connection to Master DB are allow?
- When connection to User DBs are allow?
- Delete Tempdb files?

# Why it faster?

---

- Checkpoint truncate log but not flush log to disk.
- Not everything is written to log (**minimally logged**).
- Lazy commit instated or lazy writer.



# What I feared of ?

---

- MSSQLSERVEREvent ID: 17052 Description:  
The log file for database 'TempDB' is full.Back up the transaction log for the database to free upsome log space
- Server: Msg8624, Level 16, State 1  
Internal SQL Server error
- Server: Msg1101, Level 17, State 10, Line 1  
Could not allocate new page for database 'TempDB'. There are no more pages available in filegroupDEFAULT. Space can be created by dropping objects, adding additional files, or allowing file growth.

- 
- sys.dm\_db\_file\_space\_usage
    - Returns space usage information for each file in the database
  - Spaces are in pages (8K), therefor )\*1.0/128

```
SUM(unallocated_extent_page_count) AS [free pages],  
(SUM(unallocated_extent_page_count)/128.0) AS [free space in MB]  
FROM sys.dm_db_file_space_usage;
```

- 
- `sys.dm_db_session_space_usage`
    - Returns the number of pages allocated and deallocated by each session for the database.
  - `sys.dm_db_task_space_usage`
    - Returns page allocation and deallocation activity by task for the database.

# What is so temporary in tempDB?

---

- The **tempdb** system database is a global resource that is available to all users connected to the instance of SQL Server and is used to hold the following:
- **Temporary** user **objects** that are explicitly created..
- **Internal objects** that are created by the SQL Server Database Engine...
- **Row versions**...

# Temporary user objects

---

- explicitly created:
  - global or local temporary tables, #table ##table
  - temporary stored procedures,
  - table variables,
  - cursors.
  - Tables returned in table-valued functions

# Temporary user objects

- explicitly created:

- **global** or **local** temporary tables, **#table** and **##table**

Local temporary tables are **visible only to their creators** during the **same connection** to an instance of SQL Server as when the tables were first created or referenced.

Local temporary tables are **deleted** after the user **disconnects** from the instance of SQL Server.

Global temporary tables are **visible to any user** and **any connection** after they are created,

Global temporary tables are **deleted** when **all users** that are referencing the table **disconnect** from the instance of SQL Server.

- temporary stored procedures,
- table variables,
- cursors.

# Demo 1

---

- 3 windows

# Demo 1

---

- Global temporary tables are automatically dropped when the session that created the table ends and all other tasks have stopped referencing them. The association between a task and a table is maintained only for the life of a single Transact-SQL statement. This means that a global temporary table is dropped at the completion of the last Transact-SQL statement that was actively referencing the table when the creating session ended.



# temporary stored procedures

---

- **Private** and global temporary stored procedures, analogous to temporary tables.
- with the # (Private ) and ## (global ) prefixes added to the procedure name.
- A **Local** temporary procedure is automatically deleted when the creator connection is closed..
- A **global** temporary stored procedure is automatically deleted when the creator connection is closed and there is no other connection s using it. no further execution of the global temporary stored procedure is allowed once the creator connection is closed.
- Regular stored procedure in TempDB is automatically deleted when SQL Server is shut down.
- Temp SP vs. [sp\\_executeSQL](#)?
- **sp\_executesql** does not store data in the system tables.

[https://technet.microsoft.com/en-us/library/ms190669\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms190669(v=sql.105).aspx)



# table variables

---

- Like other variables are declared in the body of a batch or procedure with the DECLARE statement.
- Constraint types allowed are PRIMARY KEY, UNIQUE, NULL, and CHECK.
- Location?
- Transaction?
- Log?
- Lifetime?
- Visibility to different scopes?
- Metadata? - Demo
- statistics & query plans - trace flag 2453 (SS2k12SP2)
- Locking - Demo

- 
- Indexes- before ss2K14 and after
  - Parallelism?
  - Char collation?

# Demo 2 table variables - Metadata

---

# Update statistics

---

- Based on count of the number of modifications made to each column.
- If number modifications since the plan was compiled exceeds recompilation threshold (RT) then Update statistics and recompiled

# Update statistics

---

- Assume  $n$  refers to a table's cardinality when a query plan is compiled:
- **Permanent table**
  - If  $n \leq 500$ ,  $RT = 500$ .
  - If  $n > 500$ ,  $RT = 500 + 0.20 * n$ .
- **Temporary table**
  - If  $n < 6$ ,  $RT = 6$ .
  - If  $6 \leq n \leq 500$ ,  $RT = 500$ .
  - If  $n > 500$ ,  $RT = 500 + 0.20 * n$ .
- **Table variable**
  - RT does not exist. Therefore, recompilations do not happen

# Demo 3 - table variables - Locking

---

- Insert vs. Select

# cursors

---

- Keyset-driven cursors store the set of keys that identify the rows in the cursor.
- Static cursors use a work table to store the complete result set of the cursor.



# Internal objects

---

- Work tables for cursor
- **spool** operations (missing memory)
- Work files for operations such as a hash join or hash aggregate operations.
- Intermediate sort results like:
  - Online Index create or rebuild operations
  - GROUP BY, ORDER BY, or UNION queries.
- Common table expression
- DBCC CHECKDB work tables
- temporary large object (LOB) storage
- Service Broker , Database Mail , Event notifications

# Demo 4 **spool** operations

---

- <http://www.sqlskills.com/InsiderContent/YouTube/Insider201110Code2.zip>

# Large object (LOB) data type

---

- `varchar(max)`, `nvarchar(max)`, `varbinary(max)`, `text`, `ntext`, `image`, and `xml`.
- Up to 2 GB in size
- Can be used as variables or parameters in stored procedures, user-defined functions, batches, or queries.
- Parameters and variables that are defined as a LOB data type use main memory as storage if the values are small. However, large values are stored in **tempdb**.

# Service Broker

---

- preserving existing dialog context that cannot stay in memory.
- The size is approximately 1 KB per dialog.
- Database Mail, Event Notifications, and Query Notifications implicitly use Service Broker.

# Row versioning

---

- Triggers (**inserted** and **deleted** tables )
- Bulkload operations with triggers enabled
- MARS (Multiple Active Result sets)
- Index operations that specify the ONLINE option
- Always On Availability Group
- Read-committed isolation level that uses row versioning to provide statement-level read consistency.
- Snapshot isolation level to provide transaction-level read consistency.

# Row versioning

---

- Held in the **tempdb** version store for as long as an active transaction must access it.
- sys.dm\_tran\_version\_store
- **version\_store\_reserved\_page\_count** column in sys.dm\_db\_file\_space\_usage
- An active transaction must keep all its changes in the version store.
- $$[\text{Size of Version Store}] = 2 * [\text{Version store data generated per minute}] * [\text{Longest running time (minutes) of your transaction}]$$

# Multiple Active Result Sets (MARS)

---

- If a MARS session issues a data modification statement when there is an active result set, the rows that are affected by the modification statement are stored in the version store in **tempdb**.

# TEmpDB SS2014

---

- Speed Up select into #Table
- Eager writes reduce the pressure on lazy writer and checkpoint by tracking the dirtied bulk operations pages in a circular list.
- In SQL Server 2014, **Create index WITH SORT IN TEMPDB** or **select into #Table** , SQL Server now recognize this may be a short lived operation, and the SQL Server 2014, eager write behavior, no longer forces these pages to disk as quickly as previous versions.



# User Group 11-2015

---

## Back to Basics – Locks Behind the Scenes Part 1

Dubi Lebel  
Dubi Or Not To Be

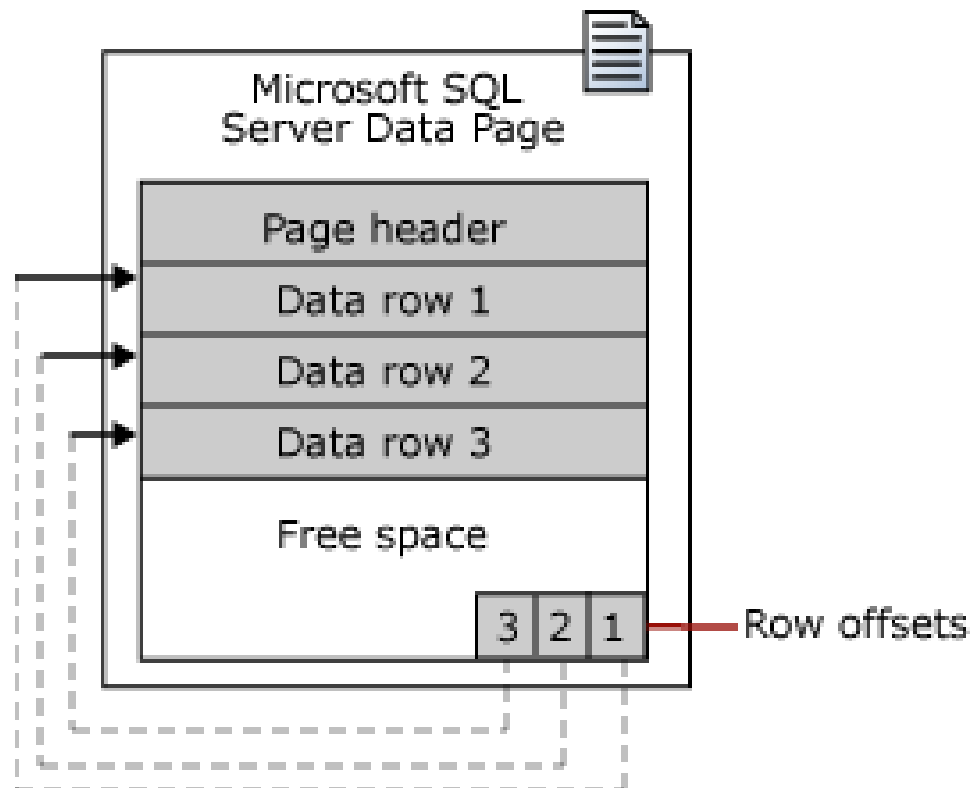
Latches in TempDB

# B2B

---

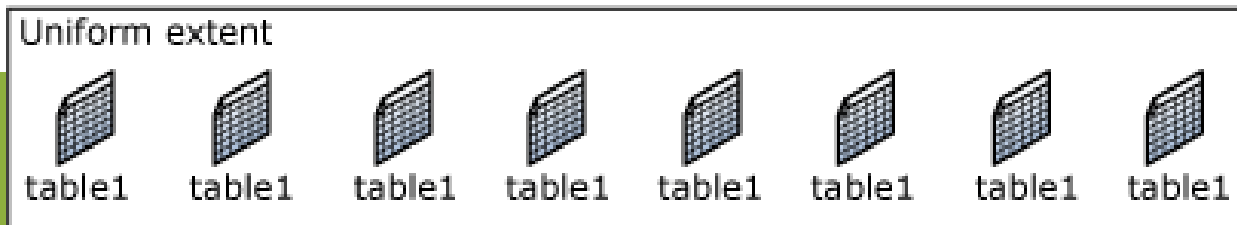
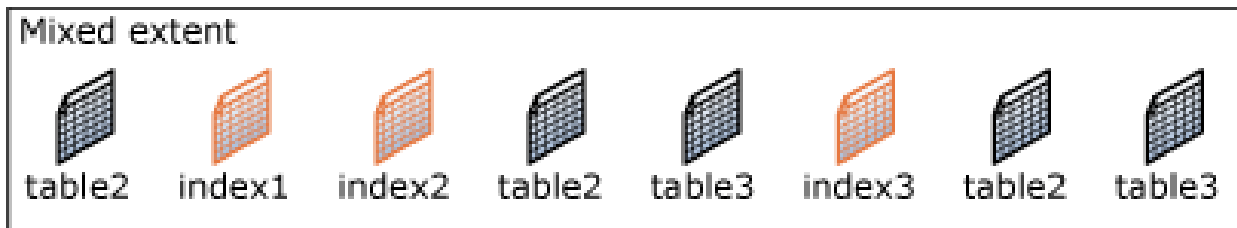
- [https://technet.microsoft.com/en-us/library/ms190969\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms190969(v=sql.105).aspx)

# Page 8 KB



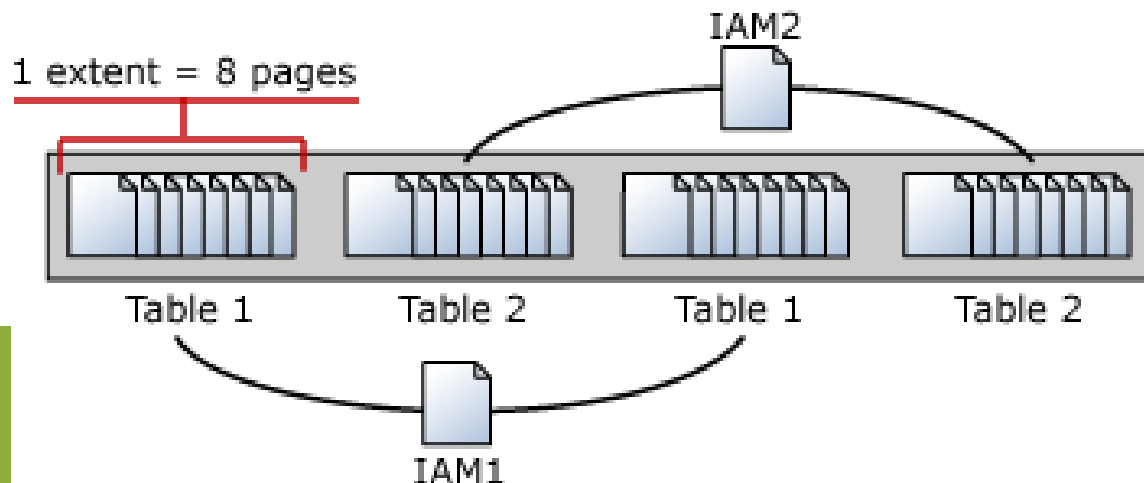
# Extent

- Extents are the basic unit in which space is managed
- An extent is eight physically contiguous pages (64 KB)
- 16 extents per megabyte.
- Uniform extents are owned by a single object.
- Mixed extents are shared by up to eight objects.



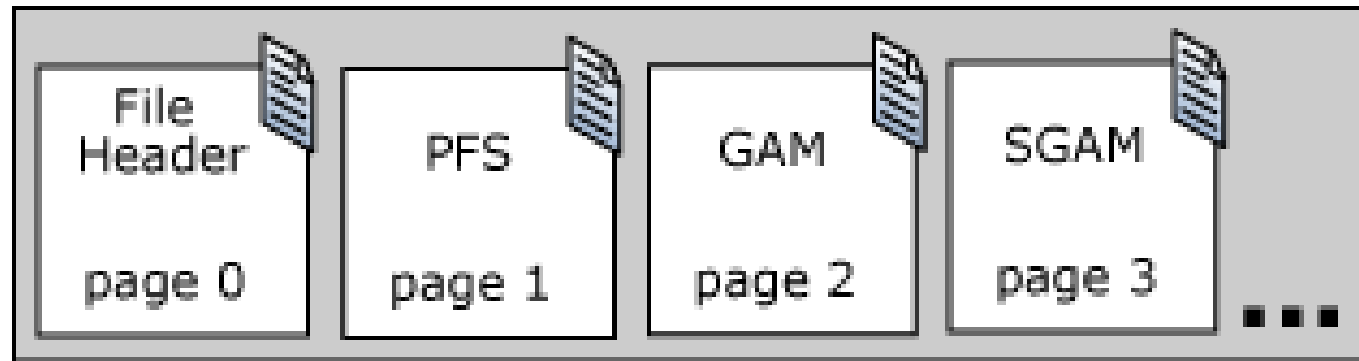
# TempDB page allocation

- Each **Temporary** user **objects** uses a minimum of **two (2)** one IAM page and one Data-page – in a mixed extent.
- Each **internal** object uses a minimum of **nine (9)** pages; one IAM page and one eight-page - extent.



# First Data file pages

---



# Page Free Space (PFS)

---

- PFS page is the first page after the file header page in a data file.
- Record the allocation status of each page.
- Has one byte for each page, recording whether the page is allocated, and if so, whether it is empty, 1 to 50 percent full, 51 to 80 percent full, 81 to 95 percent full, or 96 to 100 percent full.
- Next PFS on multiple 8088

# Global Allocation Map (GAM)

---

- Page 2 of each database is the GAM page
- GAM pages record what extents have been allocated.
- Each GAM covers 64,000 extents, or almost 4 GB of data.
- next GAM on multiple 511,232
- The GAM has one bit for each extent in the interval it covers.
- If the bit is 1, the extent is free; if the bit is 0, the extent is allocated.



# Shared Global Allocation Map (SGAM)

---

- SGAM pages record which extents are currently being used as mixed extents and also have at least one unused page.
- Each SGAM covers 64,000 extents, or almost 4 GB of data.
- next SGAM on multiple 511,232
- The SGAM has one bit for each extent in the interval it covers. If the bit is 1, the extent is being used as a mixed extent and has a free page.
- If the bit is 0, the extent is not used as a mixed extent, or it is a mixed extent and all its pages are being used.

# For the curious

---

- What are the next pages?



# TempDB Latch Contention

---

- Contention on System data pages PFS, GAM, SGAM

# Tempdb number of files

---

- As a general rule, if the number of logical processors **is less than or equal to 8**, **use the same number of data files as logical processors**.
- If the number of logical processors is **greater than 8**, **use 8 data files** and then if contention continues, increase the number of data files by **multiples of 4** (up to the number of logical processors) until the contention is reduced to acceptable levels or make changes to the workload/code.
- <https://support.microsoft.com/en-us/kb/2154845>

# Uneven files

---

- SQL Server use propositional fill.
- if files are not in the same size it will use the file with more empty space

# Trace Flag

---

- KB 2154845
- Trace 1118 - avoid “mixed extents” and use “full extents”.
- Trace 1117 - Forces all file under file group to grow equally
- Trace 1106 - SQL 9 - Used space in tempdb increases continuously when you run a query that creates internal objects in tempdb. KB 947204.

# Demo TempDB Latch Contention

---

---

# Do we have solved all the problems?

- Adding more Data files
- Trace flags 1117 & 1118 are on



# Do we have solved all the problems? **NO**

---

- Adding more Data files
- Trace flags 1117 & 1118 are on
- **Monitor** Tempdb file size. It could happened that one file will grow disproportionately.
- TF 1118 solved only the SGAM Contention, what about FPS and GAM?

# SS16

## Database Engine Configuration

Specify Database Engine authentication security mode, administrators, data directories and TempDB settings.

Global Rules

Product Updates

Install Setup Files

Install Rules

Installation Type

Product Key

License Terms

Setup Role

Feature Selection

Feature Rules

Instance Configuration

Server Configuration

**Database Engine Configuration**

Feature Configuration Rules

Ready to Install

Installation Progress

Complete

Server Configuration

Data Directories

TempDB

FILESTREAM

TempDB data files: tempdb.mdf, tempdb\_mssql\_#.ndf

Number of files:

8

Initial size (MB):

8

Total initial size (MB):

64

Autogrowth (MB):

64

Total autogrowth (MB):

512

Data directories:

C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER

Add...

Remove

TempDB log file:

templog.ldf

Initial size (MB):

8

Autogrowth (MB):

64

Log directory:

C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER

...

< Back

Next >

Cancel

Help

- 
- **TF 1117 and 1118 is enabled by default for tempdb**
  - **Data Files:**
    - **Number of files – 8?**
    - **Initial size** In MB applies to each tempdb data file
    - **Autogrowth** in MB, default 64MB to cover one PFS interval.
    - **Data directories** could spread the data files across several volumes.
  - **Log File**
    - **Initial size** In MB
    - **Autogrowth** in MB, default 64MB

# 1117 & 1118 on User Database

---

- replaced by a new ALTER DATABASE

- 1117 - Default value is OFF

```
ALTER DATABASE <dbname> SET  
MIXED_PAGE_ALLOCATION { ON | OFF }
```

- 1118 - Default value is AUTOGROW\_SINGLE\_FILE

```
ALTER DATABASE <dbname> MODIFY FILEGROUP  
<filegroup> { AUTOGROW_ALL_FILES |  
AUTOGROW_SINGLE_FILE }
```

# Disk response time

---

- MS recommended disk response time 3-5 ms.  
for TempDB log , 10-12 for Tempdb Data.
- Raid 10
- RAM disk
- SSD

# Conclusion: what is TempDB?

---

# Conclusion: what is TempDB

---



- 
- How to operate single toilet room in large and loud party?
  - Only one TempDB per instance.



# Be prepared



<https://skeptisys.wordpress.com/tag/kittens/>

- 
- [Working with tempdb in SQL Server 2005 Whitepaper](#)
  - [Paul Randal blog on tempdb](#)
  - [SQL Storage Engine Team posts on tempdb](#)
  - Bob Ward blog
  - MSDN

# Please fill online evaluation for both speakers and overall event.

---

You have both links in the last EVENT UPDATE email:

Session evaluation form:

<http://www.sqlsaturday.com/481/sessions/sessionevaluation.aspx>

Overall event evaluation form:

<http://www.sqlsaturday.com/###/EventEval.aspx>

# Special thanks to our great sponsors!

---

In partnership with

