# SQL Server 2012 Database & Database Files Lab

# Table of Contents

# Lab Objectives

# Virtual PC Configuration and Setup

# Lab 2

## Troubleshooting tempdb Space Issue

# Exercise 1

## Troubleshooting tempdb Space Issue by moving tempdb into a different location

## Overview

## Objectives

## Task

1. **Setup the virtual machine**.

* Start Virtual Machine
* On the SQL2012 Virtual Machine Node go to snapshot and apply SQL2012(Module 1) snapshot.
* After the snapshot is applied start SQL2012 Virtual Machine.
* Connect to SQL2012 Virtual Machine and login using below credential
* Login ID: administrator
* Password: di123!

1. **Start SQL Server and open scripts used in this LAB**

* Open SQL Server Management Studio and connect to the server using below credentials
  + Server Name: SQL2012\SQL2012SP1
  + Login: sa
  + Password: di123!
* Open TEMPDBFULL.sql located in C:\temp\module2\Labs\Lab2 folder with SQL Server Management Studio.

1. **Restore PWI\_TEMPDBFULL database**

* Open setup.sql located in C:\temp\module2\Labs\Lab2 folder with SQL Server Management Studio.
* Execute script provided in STEP 1 of setup.sql to restore PWI\_TEMPDBFULL from disk.

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

--STEP 1

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

--Execute below script to restore PWI\_TEMPDB\_FULL database from PWI\_TEMPDBFULL.bak located in C:\temp\module2\Database Backup folder.

USE [master]

IF EXISTS (SELECT name FROM sys.databases WHERE name = 'PWI\_TEMPDBFULL')

DROP DATABASE PWI\_TEMPDBFULL;

GO

RESTORE DATABASE [PWI\_TEMPDBFULL] FROM DISK = N'C:\temp\module2\Database Backup\PWI\_TEMPDBFULL.bak' WITH FILE = 1, MOVE N'PWI\_TEMPDBFULL' TO N'C:\temp\module2\Database Files\PWI\_TEMPDBFULL.mdf', MOVE N'PWI\_TEMPDBFULL\_log' TO N'C:\temp\module2\Database Files\PWI\_TEMPDBFULL\_log.ldf', NOUNLOAD, STATS = 5

GO

1. **Check current size of tempdb**

* Go to TEMPDBFULL.sql query window and execute query provided in STEP 2 to find the current size of tempdb. Note down the current size.

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

--STEP 2

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

--Execute below query to find current size of tempdb.

USE [tempdb]

GO

SELECT name AS [Logical Name], size/128.0 AS [Total Size in MB],

size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS int)/128.0 AS [Available Space In MB]

FROM sys.database\_files

--Current Size:

1. **Increase tempdb size**

* Tempdb is mainly used as a global resource for storing temporary objects.
* Mainly three types of objects can be stored in tempdb
  + Temporary user objects like local and global temp tables.
  + ROW VERSION STORE used by snapshot isolation.
  + INTERNAL OBJECTS used by SQL Server for operations like short, spool etc.
* In this lab scenario we will use temporary user object.
* Go to setup.sql query window and execute scripts provided in STEP 2 and STEP 3.

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

--STEP 2

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

--Create table tempTAB

USE [tempdb]

GO

CREATE TABLE [dbo].[tempTAB](

[SID] [bigint] NOT NULL,

[PID] [int] NOT NULL,

[FNAME] [varchar](50) NOT NULL,

[LNAME] [varchar](50) NOT NULL,

[ADDRESS] [varchar](max) NOT NULL,

[SALARY] [int] NOT NULL)

GO

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

--STEP 3

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

--Create Stored Procedure to populate data on tempTAB

USE [tempdb];

GO

CREATE PROCEDURE [dbo].[sp\_populate\_data]

@TAB AS NVARCHAR(50), @ROW AS INT

AS

SET NOCOUNT ON

DECLARE @RowCount INT

DECLARE @RowMax INT

DECLARE @RowString INT

DECLARE @SQL1 NVARCHAR(200)

SET @SQL1 = N'SELECT @RowCount1=COUNT(1) FROM '+@TAB

EXEC sp\_executesql @SQL1, N'@RowCount1 INT OUTPUT', @RowCount1 = @RowCount OUTPUT

SET @RowMax = @RowCount + @ROW

WHILE @RowCount < @RowMax

BEGIN

SET @RowString = CAST(@RowCount AS INT)

DECLARE @FNAME VARCHAR(30)

SET @FNAME = (SELECT CAST(CAST(newid() as binary(16)) as varchar(8)))

DECLARE @num1 VARCHAR(20)

SET @num1 = (SELECT CONVERT(INT, (2000+1)\*RAND()))

DECLARE @name1 VARCHAR(50)

SET @name1 = @FNAME +'\_'+ @num1;

--SELECT @name1;

DECLARE @LNAME VARCHAR(30)

SET @LNAME = (SELECT CAST(CAST(newid() as binary(16)) as varchar(8)))

DECLARE @num2 VARCHAR(20)

SET @num2 = (SELECT CONVERT(INT, (2000+1)\*RAND()))

DECLARE @name2 VARCHAR(50)

SET @name2 = @LNAME +'\_'+ @num2;

--SELECT @name2;

DECLARE @ADDRESS VARCHAR(200)

SET @ADDRESS = (SELECT CAST(CAST(newid() as binary(16)) as varchar(50)))

--SELECT @ADDRESS

DECLARE @SID INT

SET @SID = (REPLICATE('0', 10 - DATALENGTH(@RowString)) + @RowString)

--SELECT @SID;

DECLARE @PID INT

SET @PID = (SELECT CONVERT(INT, (200000+1)\*RAND()))

--SELECT @PID;

DECLARE @SALARY INT

SET @SALARY = (SELECT CONVERT(INT, (200000+1)\*RAND()))

DECLARE @SQL2 NVARCHAR(MAX)

SET @SQL2 = N'INSERT INTO '+@TAB+ N' VALUES(

@SID11,@PID11,@name11,@name22,@ADDRESS11,@SALARY11);'

DECLARE @PARAM NVARCHAR(MAX)

SET @PARAM = N'@SID11 INT,@PID11 INT,@name11 VARCHAR(50),@name22 VARCHAR(50),@ADDRESS11 VARCHAR(50),@SALARY11 INT'

EXEC sp\_executesql @SQL2, @PARAM, @SID11 = @SID, @PID11 = @PID,

@name11 = @name1, @name22 = @name2, @ADDRESS11 = @ADDRESS, @SALARY11 = @SALARY

--EXEC sp\_executesql @SQL2

SET @RowCount = @RowCount + 1

END

GO

1. **Populate data into temp table to increase tempdb size.**

* Go to TEMPDBFULL.sql query window and execute stored procedure provided in STEP 3 to populate data into tempTAB table. This will take some time to execute (Approximate 5 mins)

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

--STEP 3

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

--Open setup.sql and execute STEP 2 and STEP 3 to create table tempTAB and stored procedure sp\_populate\_data in tempdb database.

--Execute to populate data

EXEC sp\_populate\_data tempTAB, 2000000

1. **Limit tempdb MAXSIZE**

* In this step we will limit tempdb growth to a limited value. This will simulate what happens when tempdb is full and not able to grow anymore.

Execute script provided in STEP 4 to limit tempdb growth.

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

--STEP 4

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

--Execute below query to limit the tempdb size to 170 MB

USE [master]

GO

ALTER DATABASE [tempdb] MODIFY FILE ( NAME = N'tempdev', MAXSIZE = 174080KB )

GO

1. **Increase tempdb size**

* Next we will increase tempdb size again to the MAXSIZE defined in the last step. This will make tempdb full. Note the query will return an error as tempdb will be no longer able to allocate any space. Execute stored procedure provided in STEP 5.

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

--STEP 5

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

--Execute to populate data (This will make tempdb full as MAX size is 170 MB)

USE [tempdb];

EXEC sp\_populate\_data tempTAB, 200000

--Msg 1105, Level 17, State 2, Line 1

--Could not allocate space for object 'dbo.tempTAB' in database 'tempdb' because the 'PRIMARY' filegroup is full.

1. **Check current size of tempdb.**

* Execute query provided in STEP 6 to find the current size of tempdb.

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

--STEP 6

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

--Execute below query to find current size of tempdb.

USE [tempdb]

GO

SELECT name AS [Logical Name], size/128.0 AS [Total Size in MB],

size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS int)/128.0 AS [Available Space In MB]

FROM sys.database\_files

--tempdb current size:

1. **Some effects of tempdb disk space issue**

* You can execute query provided in STEP 7 to observe some errors due to tempdb full.

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

--STEP 7

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

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

--Error 1101

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

USE [PWI\_TEMPDBFULL]

GO

DBCC CHECKDB

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

--Error 1105

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

USE [tempdb];

EXEC sp\_populate\_data tempTAB, 2000

1. **Move tempdb to a different location with sufficient disk space**

* Find the current path and logical name of tempdb. This information is used in the ALTER DATABASE command in the next step to move tempdb files. Execute script in STEP 8

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

--STEP 8

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

--Note Logical Name and Path

USE [tempdb]

GO

SELECT name AS [Logical Name], physical\_name AS [PATH]

FROM sys.database\_files

--Logical Name

--Path

* Execute ALTER DATABASE MODIFY FILE command to move the tempdb files.

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

--STEP 9

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

--Execute below query to move tempdb database files in a different location.

ALTER DATABASE tempdb MODIFY FILE ( NAME = tempdev , FILENAME = 'C:\temp\module2\Database Files\tempdb.mdf' )

ALTER DATABASE tempdb MODIFY FILE ( NAME = templog , FILENAME = 'C:\temp\module2\Database Files\templog.ldf')

* Stop SQL Server Instance (SQL2012SP1) from configuration manager or using SHUTDOWN tsql provided in STEP 10.

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

--STEP 10

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

--Stop SQL Server Service from configuration manager or SSMS or Command Prompt or Using below command

--SHUTDOWN

* Start SQL Server Instance (SQL2012SP1) from configuration manager.

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

--STEP 11

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

--Start SQL Server Service from configuration manager or SSMS or Command Prompt.

* Verify the change using query provided in STEP 12

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

--STEP 12

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

--Verify changes by executing below query

USE [tempdb]

GO

SELECT name AS [Logical Name], physical\_name AS [PATH]

FROM sys.database\_files

* Delete the old tempdb files from the previous location obtained from STEP 1 and change the MAXSIZE of tempdb to UNLIMITED growth.

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

--STEP 13

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

--Delete OLD tempdb files from the PATH obtained in STEP 1.

--Execute below query to allow tempdb to grow unlimited in size

USE [master]

GO

ALTER DATABASE [tempdb] MODIFY FILE ( NAME = N'tempdev', MAXSIZE = UNLIMITED )

GO

# Lab 2

## Troubleshooting tempdb Space Issue

# Exercise 2

## Troubleshooting tempdb Space Issue by shrinking tempdb files

## Overview

## Objectives

## Task

1. **Setup the virtual machine**.

* Start Virtual Machine
* On the SQL2012 Virtual Machine Node go to snapshot and apply SQL2012(Module 1) snapshot.
* After the snapshot is applied start SQL2012 Virtual Machine.
* Connect to SQL2012 Virtual Machine and login using below credential
* Login ID: administrator
* Password: di123!

1. **Start SQL Server and open scripts used in this LAB**

* Open SQL Server Management Studio and connect to the server using below credentials
  + Server Name: SQL2012\SQL2012SP1
  + Login: sa
  + Password: di123!
* Open TEMPDBSHRINK.sql located in C:\temp\module2\Labs\Lab2 folder with SQL Server Management Studio.

1. **Restore PWI\_TEMPDBFULL database**

* Open setup.sql located in C:\temp\module2\Labs\Lab2 folder with SQL Server Management Studio.
* Execute script provided in STEP 1 of setup.sql to restore PWI\_TEMPDBFULL from disk.

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

--STEP 1

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

--Execute below script to restore PWI\_TEMPDB\_FULL database from PWI\_TEMPDBFULL.bak located in C:\temp\module2\Database Backup folder.

USE [master]

IF EXISTS (SELECT name FROM sys.databases WHERE name = 'PWI\_TEMPDBFULL')

DROP DATABASE PWI\_TEMPDBFULL;

GO

RESTORE DATABASE [PWI\_TEMPDBFULL] FROM DISK = N'C:\temp\module2\Database Backup\PWI\_TEMPDBFULL.bak' WITH FILE = 1, MOVE N'PWI\_TEMPDBFULL' TO N'C:\temp\module2\Database Files\PWI\_TEMPDBFULL.mdf', MOVE N'PWI\_TEMPDBFULL\_log' TO N'C:\temp\module2\Database Files\PWI\_TEMPDBFULL\_log.ldf', NOUNLOAD, STATS = 5

GO

1. **Check current size of tempdb**

* Go to TEMPDBSHRINK.sql query window and execute query provided in STEP 2 to find the current size of tempdb. Note down the current size.

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

--STEP 2

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

--Execute below query to find current size of tempdb.

USE [tempdb]

GO

SELECT name AS [Logical Name], size/128.0 AS [Total Size in MB],

size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS int)/128.0 AS [Available Space In MB]

FROM sys.database\_files

--Current Size:

1. **Check tempdb space usage by different objects**

* Tempdb contains three types of temporary objects
  + Temporary user objects like local and global temp tables.
  + ROW VERSION STORE used in snapshot isolation level.
  + INTERNAL OBJECT used in operations like short, spool etc.
* Execute query provided in STEP 3 to find space usage by this objects.

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

--STEP 3

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

--Execute and note the output of the below query

USE [tempdb];

GO

select getdate() AS runtime, SUM (user\_object\_reserved\_page\_count)\*8 as usr\_obj\_kb,

SUM (internal\_object\_reserved\_page\_count)\*8 as internal\_obj\_kb,

SUM (version\_store\_reserved\_page\_count)\*8 as version\_store\_kb,

SUM (unallocated\_extent\_page\_count)\*8 as freespace\_kb,

SUM (mixed\_extent\_page\_count)\*8 as mixedextent\_kb

FROM sys.dm\_db\_file\_space\_usage

--runtime usr\_obj\_kb internal\_obj\_kb version\_store\_kb freespace\_kb mixedextent\_kb

1. **Increase tempdb size**

* Increase tempdb size by creating a temp table and inserting some data in it.
* Execute script in STEP 4 to create the temp table and populate data.

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

--STEP 4

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

--Create a Global temp table

USE [PWI\_TEMPDBFULL]

GO

CREATE TABLE ##NewGlobalTempTable(

[SID] [bigint] NOT NULL,

[PID] [int] NOT NULL,

[FNAME] [varchar](50) NOT NULL,

[LNAME] [varchar](50) NOT NULL,

[ADDRESS] [varchar](max) NOT NULL,

[SALARY] [int] NOT NULL)

GO

--Open setup.sql and execute script in STEP 3

--Execute below stored procedure to populate data in ##NewGlobalTempTable table

USE tempdb;

EXEC sp\_populate\_data ##NewGlobalTempTable,100000

1. **Check tempdb space distribution**

* Again check tempdb space distribution by executing query provided in STEP 5. Note this time usr\_obj\_kb column will have a increased value as we have inserted data in temporary user table.

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

--STEP 5

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

--Execute and note the output of the below query

USE [tempdb];

GO

select getdate() AS runtime, SUM (user\_object\_reserved\_page\_count)\*8 as usr\_obj\_kb,

SUM (internal\_object\_reserved\_page\_count)\*8 as internal\_obj\_kb,

SUM (version\_store\_reserved\_page\_count)\*8 as version\_store\_kb,

SUM (unallocated\_extent\_page\_count)\*8 as freespace\_kb,

SUM (mixed\_extent\_page\_count)\*8 as mixedextent\_kb

FROM sys.dm\_db\_file\_space\_usage

--runtime usr\_obj\_kb internal\_obj\_kb version\_store\_kb freespace\_kb mixedextent\_kb

--Note: User Object (usr\_obj\_kb) is increased in size.

1. **Increase tempdb size further using ROW VERSION STORE.**

* Normally tempdb is self-maintained, means all the temporary objects are automatically cleared when the session that assigns them is over. But for this lab we will use an explicit transaction to stop this auto clearing to temporary objects. Copy the script provided in STEP 6 and execute in a new query window.

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

--STEP 6

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

--Open a new query window and execute below query

USE [PWI\_TEMPDBFULL]

GO

CREATE TABLE OpenTran (NAME VARCHAR(50));

GO

BEGIN TRAN

GO

INSERT INTO OpenTran VALUES('PWI');

GO

* Next enable READ\_COMMITTED\_SNAPSHOT isolation level by executing ALTER DATABASE command provided in STEP 7.

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

--STEP 7

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

--Enable READ COMMITED SNAPSHOT on database PWI\_TEMPDBFULL

USE [master]

GO

ALTER DATABASE [PWI\_TEMPDBFULL] SET READ\_COMMITTED\_SNAPSHOT ON WITH NO\_WAIT

GO

* Now execute UPDATE command provided in STEP 8 to update TAB1. This will generate lots of ROW VERSION as our database is in READ\_COMMITTED\_SNAPSHOT isolation level.

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

--STEP 8

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

USE [PWI\_TEMPDBFULL]

GO

UPDATE [TAB1]

SET [FNAME] = 'TEST', [LNAME] = 'TEST'

--While the above query is executing open a new query window and

--execute query in STEP 5.

--You will notice ROWVERSION (version\_store\_kb) to increase in size.

1. **Check tempdb size**

* Execute query provided in step 9 to check tempdb current size.

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

--STEP 9

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

--Execute below query to find current size of tempdb.

USE [tempdb]

GO

SELECT name AS [Logical Name], size/128.0 AS [Total Size in MB],

size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS int)/128.0 AS [Available Space In MB]

FROM sys.database\_files

1. **Troubleshooting**

* First check tempdb space distribution. You can use query provided in STEP 10 to do that.

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

--STEP 10

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

--Execute below query to find what tempdb space used for?

USE [tempdb];

GO

select getdate() AS runtime, SUM (user\_object\_reserved\_page\_count)\*8 as usr\_obj\_kb,

SUM (internal\_object\_reserved\_page\_count)\*8 as internal\_obj\_kb,

SUM (version\_store\_reserved\_page\_count)\*8 as version\_store\_kb,

SUM (unallocated\_extent\_page\_count)\*8 as freespace\_kb,

SUM (mixed\_extent\_page\_count)\*8 as mixedextent\_kb

FROM sys.dm\_db\_file\_space\_usage

--User Objects??

--Row Version Store??

* From the output of the above query it is clear that mainly ROW VERSION STORE and Temporary objects has got the maximum space allocated in tempdb.

ROW VERSION STORE generally cleared automatically unless there is an older open transaction. Execute query in STEP 11 to drop the temporary user objects.

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

--STEP 12

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

--Remove any temp user objects from tempdb

DROP TABLE ##NewGlobalTempTable;

GO

* Next we need to find any older open transaction that is causing ROW VERSION to stay in tempdb. Execute query provided in STEP 13 to find that.

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

--STEP 13

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

--Is there any open transaction?

SELECT

trans.session\_id as [SPID],

trans.transaction\_id as [Transaction ID],

tas.name as [Transaction Name],

tds.database\_id as [Database ID]

FROM sys.dm\_tran\_active\_transactions tas

INNER JOIN sys.dm\_tran\_database\_transactions tds

ON (tas.transaction\_id = tds.transaction\_id )

INNER JOIN sys.dm\_tran\_session\_transactions trans

ON (trans.transaction\_id=tas.transaction\_id)

WHERE trans.is\_user\_transaction = 1 -- user

AND tas.transaction\_state = 2 --active

AND tds.database\_transaction\_begin\_time IS NOT NULL

--Note the SPID

* Now kill the open transaction using KILL command provided in STEP 14, replacing SPID with spid you got in the previous step. And then check tempdb space distribution again. If all the temporary objects are cleared then we will be able to shrink tempdb file.

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

--STEP 14

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

KILL <SPID>

--Execute below query to find what tempdb space used for?

USE [tempdb];

GO

select getdate() AS runtime, SUM (user\_object\_reserved\_page\_count)\*8 as usr\_obj\_kb,

SUM (internal\_object\_reserved\_page\_count)\*8 as internal\_obj\_kb,

SUM (version\_store\_reserved\_page\_count)\*8 as version\_store\_kb,

SUM (unallocated\_extent\_page\_count)\*8 as freespace\_kb,

SUM (mixed\_extent\_page\_count)\*8 as mixedextent\_kb

FROM sys.dm\_db\_file\_space\_usage

1. **Shrink tempdb file**

* Execute DBCC command provided in STEP 15 to shrink tempdb to 10 MB.

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

--STEP 15

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

--Execute below DBCC command to shrink tempdb data file

DBCC SHRINKFILE('tempdev',10)

1. **Verify tempdb size**

* After shrinking verify tempdb size by executing query provided in STEP 16.

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

--STEP 16

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

--Execute below query to verify size.

USE [tempdb]

GO

SELECT name AS [Logical Name], size/128.0 AS [Total Size in MB],

size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS int)/128.0 AS [Available Space In MB]

FROM sys.database\_files

1. **Cleanup**

* After completing this lab execute script in the CLEANUP section to DROP PWI\_TEMPDBFULL database.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CLEANUP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

USE [master]

GO

DROP DATABASE [PWI\_TEMPDBFULL]

GO