

## How to Cope With Multiple Firings from a Single Threshold Crossing

### Introduction

A problem has been identified with using transaction log thresholds to initiate transaction log dumps. It has been seen that a single crossing of a threshold can start multiple instances of the procedure defined as executing when the threshold is crossed (usually the `sp_thresholdaction` system procedure – which is used to dump transaction logs). *{It was later decided that the issue was likely caused by user log caches being flushed, released and reused when the logs were hovering around a threshold.}* This has led to transaction log dump sequences being broken because the system procedure uses the same name for the transaction log dump file in each of these instances. This causes the last transaction log dump to use that same name to overwrite any and all previous transaction log dump files that used the same name.

This document describes a methodology for ensuring that multiple firings of a threshold procedure that occur when crossing the same threshold will not cause a loss of the transaction log sequence. As a side effect, the number of transaction log dumps waiting at any one time may also be reduced.

### Observations

It has been seen that when multiple firings of `sp_thresholdaction` occur when crossing the same threshold, the file names generated by `sp_thresholdaction` when using seconds as the smallest unit of time are the same for each instance of `sp_thresholdaction` initiated by the threshold fired at that time. By using milliseconds as the smallest unit of time in the file name in conjunction with the proposal below, this problem should not occur.

Between two and four firings have been seen as being initiated by the same crossing of a threshold, resulting in up to four transaction log dumps using the same file name.

### Proposal

To ensure that the same filename is not generated by a threshold firing several instances of `sp_thresholdaction`, an extra step will be taken to ensure uniqueness. This step is to delay the procedure at the start of its execution, by a random amount of milliseconds to give a delay time between zero to just under six seconds. (This will only be performed if the transaction logs are to be dumped to a file. Truncating transaction logs does not cause a problem.) This will be accomplished using the “waitfor delay” command with the time generated by the random number function. After the delay is over, the current date and time will be determined and this will be used to generate the file name to dump the transaction logs to.

After the delay, checks will be made to determine if this instance of `sp_thresholdaction` should proceed with an attempt to dump the transaction logs. This requires that some flagging information be stored and available to other instances of `sp_thresholdaction`. As the writing of any information to a table that is accessible by all instances that may have been fired at the same time is susceptible to log or log filling problems, user tables will not be used for this flagging. Instead, the virtual table `master.dbo.sysprocesses` will be used, by utilising three new set commands introduced in ASE 12.0.

When a threshold is fired, the stored procedure that is executed is passed the following parameters

@dbname

@segment\_name

@space\_left

@status (1 if last chance threshold, otherwise 0)

To determine if any other instances of sp\_thresholdaction were fired by the same threshold, the procedure will interrogate the sysprocesses table to determine if any threshold procedures are shown as executing with :-

1. the same value as this database's ID in the dbid column
2. the same value as this procedure's ID in the id column
3. the same value as this segment's name in the clientname column
4. the same space left in clienthostname column
5. an execution time in the clientapptime column within 10 seconds of the execution time of this execution of sp\_thresholdaction

Matches for all five checks will show that the threshold was fired for multiple instances of the threshold procedure and that one has already passed this checking. Consequently, this procedure will exit.

Using the "set clientname", "set clienthostname" and "set clientapptime" commands, information about this instance of the threshold procedure will be written to the sysprocesses table. The segment name will be stored using "set clientname", the space left will be stored using "set clienthostname" and the date and time the threshold procedure was fired (determined as the very first command executed by the procedure) will be stored using "set clientapptime". (The values have to be converted to varchar (30) to allow them to be stored in the columns in the table.)

A check will then be performed to determine if there are any transaction log dumps for the database currently taking place. If no transaction log dumps are currently taking place for the database, one can be started. If only one transaction log dump is currently taking place for the database, a second transaction log dump can be started. However, this second transaction log dump will be blocked by the one that is currently taking place for the database. If there is already a transaction log dump queued behind a transaction log dump that is currently taking place, the threshold procedure will exit.

The procedure will allow one additional transaction log dump to be queued to endeavour to ensure that as much of the transaction logs are cleared down as possible. If a transaction log dump that is currently executing passes the point in the logs where the oldest open transaction record is located before that oldest open transaction is closed, the space released by the closing of that transaction will not be cleared at the end of the currently executing transaction log dump. Setting off another transaction log dump immediately after the currently executing dump completes will give the greatest opportunity of reclaiming space in the transaction logs.

*{The next section on compression is no longer germane, as checking for the flag table caused problems in some cases.}*

The file name to use to store the transaction log dumps in (including milliseconds) will be generated from an indication of whether the transaction log dumps should be compressed and from the time determined as close as possible to the point of dumping.

Compression of the transaction log dumps will be indicated by the existence of a flag table named DO\_COMPRESSED\_TRAN\_DUMP in the database being dumped. This table will not be interrogated directly. Its existence will be determined by querying the sysobjects table in the current database. If the

table exists, the compression flag will be used when the dump file name is generated. This way, no errors are generated if the table does not exist.

## Conclusion

By delaying the start of a transaction log dump by a short interval and using milliseconds as the smallest unit of time in the file name used to dump the transaction logs to, the generation of duplicate file names should no longer occur. Allowing each database to configure whether their transaction log dumps will be compressed will give the greatest flexibility of performance over speed. Checking for currently executing transaction log dumps for the database before starting a new transaction log dump will ensure that excessive numbers of transaction log dumps do not take place.

## Example 1

In the following example, a threshold at 677880 free pages fires three threshold action procedures, all with the same parameters. Number one sleeps for 2.567 seconds, number two sleeps for 1.789 seconds and number three sleeps for 2.234 seconds. When two wakes, it determines that no other threshold procedures that may have been fired at the same time are currently running; and as there are no other dumps running, a transaction log dump is started. When three wakes, it determines that a threshold procedure executed at the same time has reached the stage of identifying itself, so three exits. When one wakes, it determines that a threshold procedure executed at the same time has reached the stage of identifying itself, so one exits.

```

Threshold at 677880 free pages is crossed

sp_thresholdaction fired          sp_thresholdaction fired          sp_thresholdaction fired
@dbname = "db1",                  @dbname = "db1",                  @dbname = "db1",
@segment_name = "logsegment",    @segment_name = "logsegment",    @segment_name = "logsegment",
@space_left = 677880,            @space_left = 677880,            @space_left = 677880,
@status = 0                      @status = 0                      @status = 0
-----
@start = "16/10/04 09:06:07.234", @start = "16/10/04 09:06:07.234", @start = "16/10/04 09:06:07.234",
@delay = 2.567                   @delay = 1.789                   @delay = 2.234
-----
waitfor delay @delay             waitfor delay @delay             waitfor delay @delay
-----
                                (Same details in sysprocesses?)
                                NO
                                Put details in sysprocesses
                                (Same details in sysprocesses?)
(Same details in sysprocesses?) YES
YES                               (Dump tran running for db1?)
-----                               NO
return                               return
-----
                                @now_file = "16/10/04 09:06:09.035"
                                Dump tran db1
                                return

```

## Example 2

In the following example, the same conditions as in Example 1 are met for firings one and three. When firing two determines that it is the only / first firing, it finds that there is a dump transaction currently running for the same database and that there is also one queued but blocked for the same database, so the procedure exits. (If there had not been a queued dump, a second transaction log dump would have been executed and this would be blocked until the currently executing dump completed.)

Threshold at 677880 free pages is crossed

<u>sp_thresholdaction fired</u> <u>@dbname = "db1",</u> <u>@segment_name = "logsegment",</u> <u>@space_left = 677880,</u> <u>@status = 0</u>	<u>sp_thresholdaction fired</u> <u>@dbname = "db1",</u> <u>@segment_name = "logsegment",</u> <u>@space_left = 677880,</u> <u>@status = 0</u>	<u>sp_thresholdaction fired</u> <u>@dbname = "db1",</u> <u>@segment_name = "logsegment",</u> <u>@space_left = 677880,</u> <u>@status = 0</u>
<u>@start = "16/10/04 09:06:07.234",</u> <u>@delay = 2.567</u>	<u>@start = "16/10/04 09:06:07.234",</u> <u>@delay = 1.789</u>	<u>@start = "16/10/04 09:06:07.234",</u> <u>@delay = 2.234</u>
<u>waitfor delay @delay</u>	<u>waitfor delay @delay</u>	<u>waitfor delay @delay</u>
	(Same details in sysprocesses?) NO	
	Put details in sysprocesses	(Same details in sysprocesses?) YES
<u>(Same details in sysprocesses?)</u> YES	<u>(Dump tran running for db1?)</u> YES	<u>return</u>
<u>return</u>	(Dump tran queued for db1?) YES	
	return	