

## Oracle log switching sequence

Q: How does Oracle determines which log group to use next at the onset of log switch ?

Q: How does adding a log group alters the sequence ?

Log group to switch, is determined from controlfile. Let's review the control file for pertinent information. Dumping the control file using following commands reveals more information:

```
oradebug setmypid
```

```
oradebug dump controlf 12
```

Above commands generated a trace file in user\_dump\_dest directory and following is the excerpt from the file:

Only few lines shown to improve readability:

```
*****
LOG FILE RECORDS
*****
(blkno = 0x9, size = 72, max = 16, in-use = 8, last-recid= 15)
LOG FILE #1:
  siz: 0xc8000 seq: 0x0000022b hws: 0x2 bsz: 512 nab: 0x9 flg: 0x1 dup: 1
LOG FILE #2:
  siz: 0xc8000 seq: 0x0000022e hws: 0x2 bsz: 512 nab: 0x39 flg: 0x1 dup: 1
LOG FILE #3:
  siz: 0xc8000 seq: 0x00000230 hws: 0x1 bsz: 512 nab: 0xffffffff flg: 0x8 dup: 1
LOG FILE #4:
  siz: 0x5000 seq: 0x00000228 hws: 0x2 bsz: 512 nab: 0xc flg: 0x1 dup: 1
LOG FILE #5:
  siz: 0x5000 seq: 0x0000022a hws: 0x2 bsz: 512 nab: 0x4 flg: 0x1 dup: 1
LOG FILE #6:
  siz: 0x5000 seq: 0x0000022c hws: 0x2 bsz: 512 nab: 0x98 flg: 0x1 dup: 1
LOG FILE #7:
  siz: 0x5000 seq: 0x0000022f hws: 0x2 bsz: 512 nab: 0xe9 flg: 0x1 dup: 1
LOG FILE #8:
  siz: 0x5000 seq: 0x0000022d hws: 0x2 bsz: 512 nab: 0x5 flg: 0x1 dup: 1
```

Sequence# above is log sequence # in hexadecimal.

Oracle determines the next log group to switch:

- i) searching for the log group with lowest sequence# that is archived.
- ii) If a new log group is added, then that new log group will have log sequence# as zero and so that group will be selected
- iii) If there are two new log groups, then first one encountered in the above array is selected.

Third point is quite important, since if we drop a log group, then that log group is simply marked as deleted, not completely removed from the array. So, if the same log group is reread, then deleted entry is reused.

Luckily, above controlfile array is externalized as x\$kccl table. Instead of dumping the controlfile, we could use x\$kccl to test our understanding:

Following SQL, can predict which log group will be selected by Oracle, during next log switch:

```

SELECT   indx, lenum group#, lseq sequence#, archived,
         DECODE (lflg, 8, 'CURRENT') curstatus,
         CASE WHEN lseq = minseq AND indx = minind THEN 'NEXT'
         END NEXT
FROM (SELECT indx, lenum, lseq, lflg,
         DECODE (BITAND (lflg, 1), 0, 'NO', 'YES') archived,
         FIRST_VALUE (lseq) OVER (ORDER BY lseq, indx, BITAND (lflg, 1) DESC) minseq,
         FIRST_VALUE (indx) OVER (ORDER BY lseq, indx, BITAND (lflg, 1) DESC) minind
      FROM x$kccl
      WHERE lesiz != 0 AND inst_id = USERENV ('instance'))
ORDER BY indx

```

Script: print\_next\_log.sql

INDX	GROUP#	SEQUENCE#	ARC	CURSTAT	NEXT
0	1	563	YES		NEXT
1	2	566	YES		
2	3	570	NO	CURRENT	
3	4	568	YES		
4	5	569	YES		
5	6	564	YES		
6	7	567	YES		
7	8	565	YES		

Let's switch once and make sure that script is working :

```

alter system switch logfile;
@print_next_log

```

INDX	GROUP#	SEQUENCE#	ARC	CURSTAT	NEXT
0	1	563	NO	CURRENT	
1	2	566	YES		
2	3	570	NO		
3	4	568	YES		
4	5	569	YES		
5	6	564	YES		NEXT
6	7	567	YES		
7	8	565	YES		

-- Script correctly shows that next log switch will be to log group 1.

-- Let's add a new log group

```

alter database add logfile group 9 ('/db/test/d001/test_09.dbf') size 10M;

```

-- Let's drop a log group too

```

alter database drop logfile group 4;

```

```
@print_next_log
```

INDX	GROUP#	SEQUENCE#	ARC	CURSTAT	NEXT
0	1	571	NO	CURRENT	
1	2	566	YES		
2	3	570	YES		
4	5	569	YES		
5	6	564	YES		
6	7	567	YES		
7	8	565	YES		
8	9	0	YES		NEXT

-- Now, script shows that log will switch to group 9.

-- Let's add the log file we dropped and reexecute the script.

```

alter database add logfile group 4 ('/db/test/d001/test_04.dbf') size 10M;

```

```
@pri nt_next_l og
```

INDX	GROUP#	SEQUENCE#	ARC	CURSTAT	NEXT
0	1	571	NO	CURRENT	
1	2	566	YES		
2	3	570	YES		
3	4	0	YES		NEXT
4	5	569	YES		
5	6	564	YES		
6	7	567	YES		
7	8	565	YES		
8	9	0	YES		

We can see from above that dropping a group does not delete completely from the control file, rather marked as dropped. Adding the same group, reuses the entry. Even though, we added group 9 first, Oracle will use group 4, since that is the first entry with lowest log sequence#.

```
alter system switch logfile;
```

```
@pri nt_next_l og
```

INDX	GROUP#	SEQUENCE#	ARC	CURSTAT	NEXT
0	1	571	YES		
1	2	566	YES		
2	3	570	YES		
3	4	572	NO	CURRENT	
4	5	569	YES		
5	6	564	YES		
6	7	567	YES		
7	8	565	YES		
8	9	0	YES		NEXT

Clearly, script is working as intended and confirms our understanding.