

Overcoming ORA-01341: LOGMINER OUT-OF-MEMORY in Oracle Streams Downstream Capture

It is possible to encounter “out-of-memory” errors in a poorly configured Streams environment, even with Oracle 11g’s Automatic Memory Management feature enabled. This document describes the error seen in the database instance alert log and associated trace files and offers a solution.

Error seen in database instance alert log:

```
krvxrpt: Errors detected in process 65, role builder.
krvxmrs: Leaving by exception: 1341
ORA-01341: LogMiner out-of-memory
LOGMINER: session#=42, builder MS01 pid=65 OS id=29684 sid=1018
stopped
```

... also

```
Streams CAPTURE CP01 for ##### with pid=62, OS id=29652 stopped
ORA-01280: Fatal LogMiner Error.
```

Logminer Builder process in trace file:

```
*** 2009-08-13 08:05:32.712
*** SESSION ID:(1037.9) 2009-08-13 08:05:32.712
*** CLIENT ID:() 2009-08-13 08:05:32.712
*** SERVICE NAME:(SYS$USERS) 2009-08-13 08:05:32.712
*** MODULE NAME:(STREAMS) 2009-08-13 08:05:32.712
*** ACTION NAME:(Logminer Builder) 2009-08-13 08:05:32.712
```

```
Spill: can not find enough to spill. amountNeeded: 1993904
Session MaxMem 10485760, CacheSize 129264, MemSize 129264
```

Streams Process Initialisation Parameters:

The SQL below, executed as SYSDBA, returns a complete list of Streams initialisation parameters.

```
select decode(process_type,1,'APPLY',2,'CAPTURE') process_name,
name, value
from sys.streams$_process_params
order by 1,2
```

PROCESS_NAME	NAME	VALUE
APPLY	ALLOW_DUPLICATE_ROWS	N
APPLY	COMMIT_SERIALIZATION	FULL
APPLY	DISABLE_ON_ERROR	N
APPLY	DISABLE_ON_LIMIT	N
APPLY	MAXIMUM_SCN	INFINITE
APPLY	PARALLELISM	4
APPLY	PRESERVE_ENCRYPTION	Y
APPLY	RTRIM_ON_IMPLICIT_CONVERSION	Y
APPLY	STARTUP_SECONDS	0
APPLY	TIME_LIMIT	INFINITE
APPLY	TRACE_LEVEL	0
APPLY	TRANSACTION_LIMIT	INFINITE

APPLY	TXN_LCR_SPILL_THRESHOLD	1000000
APPLY	WRITE_ALERT_LOG	Y
APPLY	_APPLY_SAFETY_LEVEL	1
APPLY	_CMPKEY_ONLY	N
APPLY	_COMMIT_SERIALIZATION_PERIOD	0
APPLY	_DATA_LAYER	Y
APPLY	_DYNAMIC_STMTS	Y
APPLY	_HASH_TABLE_SIZE	10000000
APPLY	_IGNORE_CONSTRAINTS	NO
APPLY	_IGNORE_TRANSACTION	
APPLY	_KGL_CACHE_SIZE	100
APPLY	_MIN_USER_AGENTS	0
APPLY	_PARTITION_SIZE	10000
APPLY	_RECORD_LWM_INTERVAL	1
APPLY	_RESTRICT_ALL_REF_CONS	Y
APPLY	_SGA_SIZE	4
APPLY	_TXN_BUFFER_SIZE	320
APPLY	_XML_SCHEMA_USE_TABLE_OWNER	Y
CAPTURE	DISABLE_ON_LIMIT	N
CAPTURE	DOWNSTREAM_REAL_TIME_MINE	N
CAPTURE	MAXIMUM_SCN	INFINITE
CAPTURE	MESSAGE_LIMIT	INFINITE
CAPTURE	MESSAGE_TRACKING_FREQUENCY	2000000
CAPTURE	PARALLELISM	1
CAPTURE	SKIP_AUTOFILTERED_TABLE_DDL	Y
CAPTURE	STARTUP_SECONDS	0
CAPTURE	TIME_LIMIT	INFINITE
CAPTURE	TRACE_LEVEL	0
CAPTURE	WRITE_ALERT_LOG	Y
CAPTURE	_APPLY_BUFFER_ENTRIES	10000
CAPTURE	_APPLY_UNRESPONSIVE_SECS	300
CAPTURE	_CHECKPOINTS_PER_DAY	4
CAPTURE	_CHECKPOINT_FORCE	N
CAPTURE	_CHECKPOINT_FREQUENCY	1000
CAPTURE	_CKPT_FORCE_FREQ	1800
CAPTURE	_CKPT_RETENTION_CHECK_FREQ	21600
CAPTURE	_DIRECT_APPLY	AUTO
CAPTURE	_DISABLE_PGAHC	N
CAPTURE	_FLUSH_TIMEOUT	2
CAPTURE	_IGNORE_TRANSACTION	
CAPTURE	_IGNORE_UNSUPERR_TABLE	
CAPTURE	_LOGMINER_IDLE_READ_POLL_FREQ	500
CAPTURE	_MIN_DAYS_KEEP_ALL_CKPTS	1
CAPTURE	_SEND_STREAMS_DICTIONARY	0
CAPTURE	_SGA_SIZE	10
CAPTURE	_SKIP_LCR_FOR_ASSERT	
CAPTURE	_TURN_OFF_LIMIT_READ	N

Solution

Metalink Note: 335516.1 recommends the following parameter settings for the downstream capture process:

- 11g: parallelism=1 is the recommended setting and is the default ;

Generally, this parameter should not be modified. The only occasions where it is valid to change the value of `_SGA_SIZE` for the Capture/logminer session are under circumstances:

- ORA-1341 is observed; or
- Where there is log miner spill

However, I have witnessed Logminer happily mining logs that had previously caused the ORA-01341 after setting `PARALLELISM=4`. This is largely due to Oracle allocating the value of `_SGA_SIZE` for each Logminer Preparer process.

E.g.

```
BEGIN
dbms_capture_adm.set_parameter( capture_name => '<CAPTURE_NAME>',
                               parameter   => 'PARALLELISM',
                               VALUE      => '4' );
END;
/
```

In fact setting the parameter had a positive impact on the Streams performance.

In addition to this, the `_SGA_SIZE` “underscore” parameter for the capture process has a default of 10M, which appears to be very low. I recommend setting this parameter to 100M, particularly when replicating a high volume of transactions.



Tip:

The `streams_pool_size` database initialisation parameter may have to be increased to accommodate the increase in `_SGA_SIZE` if Automatic Memory Management (AMM) is disabled. When enabling AMM, it is good practice to set a minimum size for the `streams_pool_size`. I.e. Greater than 10% of `shared_pool_size` (which is the default).

For the capture process, the `_SGA_SIZE` parameter controls the size of the LCR cache. For large transactions i.e. bulk inserts/updates, Logminer generates a LCR for each row in the target table.

E.g.

```
BEGIN
dbms_capture_adm.set_parameter( capture_name => '<CAPTURE_NAME>',
                               parameter   => '_SGA_SIZE',
                               VALUE      => '100' );
END;
/
```