**In this Document**

**APPLIES TO:**

Oracle GoldenGate - Version 11.1.1.0.0 to 12.1.3.0.0 [Release 11.1.1 to 12.1]  
Oracle Database - Enterprise Edition - Version 8.1.7.4 to 12.1.0.2 [Release 8.1.7 to 12.1]  
Oracle Database Cloud Schema Service - Version N/A and later  
Oracle Database Exadata Cloud Machine - Version N/A and later  
Oracle Database Exadata Express Cloud Service - Version N/A and later  
Information in this document applies to any platform.

**PURPOSE**

The Oracle documentation for the ALTER TABLE ... ADD SUPPLEMENTAL LOG ...  states the following:

**supplemental\_id\_key\_clause**

Use this clause to specify that all or a combination of the primary key, unique key, and foreign key columns should be supplementally logged.

and:

* If you specify ALL COLUMNS, then the database includes in the redo log all the fixed-length maximum size columns of that row. Such a redo log is a system-generated unconditional log group.

where it has the following syntax:

DATA  
( { ALL | PRIMARY KEY | UNIQUE | FOREIGN KEY }  
    [, { ALL | PRIMARY KEY | UNIQUE | FOREIGN KEY } ]...  
)  
COLUMNS

However, the **DATA (ALL) COLUMNS** option may be confusing.  This is because it is included in the "...**key\_clause**", implying it applies to KEY columns only, whereas it actually means all the columns in the table.

So, the purpose of this document is to show the effects of the Supplemental Logging DATA (ALL) COLUMNS option, including the Trail file records extracted by GoldenGate.

**TROUBLESHOOTING STEPS**

**PREAMBLE**

This demonstration uses the Oracle database SCOTT schema, and the SCOTT.EMP table.  
Please note that;

* The Primary Key for SCOTT.EMP is column EMPNO,
* Only column ENAME is updated.

Three progressive levels of Supplemental Logging are used;

1. Database level only
2. Primary Keys columns
3. All Columns

Then, the LogMiner and GoldenGate LogDump utilities are used to show the contents of the Archived log and Trail files, respectively.

**SUMMARY OF RESULTS**

The following is primarily the SQL executed and the output from the LogMiner and LogDump utilities.  
For more details, please see the [TEST CASE DETAILS](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=191858855383153&parent=EXTERNAL_SEARCH&sourceId=TROUBLESHOOTING&id=2067514.1&_afrWindowMode=0&_adf.ctrl-state=147fy3txw5_4#TestCaseDetails) section.

***Supplemental logging at the Oracle database level only***

**Oracle database command  
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;**

**GoldenGate TRANDATA equivalent**None, it is outside the scope of the Capture role.   
As per the GoldenGate Oracle Installation and Setup Guide, the following command must be issued on the Oracle database: ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;  
Then, either ADD TRANDATA or ADD SCHEMATRANDATA commands must be issued for GoldenGate to be able to capture table data (columns).

**SQL Executed**update SCOTT.EMP set ENAME='**DBSUPLOG**' where EMPNO=7934;

**LogMiner V$LOGMNR\_CONTENTS**  
SQL\_REDO  
--------------------------------------------------------------------------------  
update "SCOTT"."EMP" set "ENAME" = '**DBSUPLOG**' where "ENAME" = 'MILLER' and ROWID = 'AAASZHAAEAAAACXAAN';

**GoldenGate LogDump**

2015/10/15 13:02:47.000.000 FieldComp            Len    30 RBA 1025

Name: SCOTT.EMP

After  Image:                                             Partition 4   G  s

 0000 000a ffff 0000 0000 0000 0000 0001 000c 0000 | ....................

 0008 4442 5355 504c 4f47                          | ..DBSUPLOG

Column     0 (x0000), Len    10 (x000a)

 ffff 0000 0000 0000 0000                          | ..........

Column     1 (x0001), Len    12 (x000c)

 0000 0008 4442 5355 504c 4f47                     | ....**DBSUPLOG**

NOTE: The Primary Key value 7934 is not included in the Trail file record i.e. Column 0 is empty.  This is because neither the ADD TRANDATA nor the ADD SCHEMATRANDATA command was issued, where this can then cause problems on the REPLICAT side.  Column 0 EMPNO value DEC 7934 = HEX 1EFE, and is not in the Trail record.

***With Primary Key columns***

**Oracle database command  
alter table SCOTT.EMP add supplemental log data (primary key) columns;**

**GoldenGate TRANDATA equivalent**add trandata scott.emp

**SQL executed**update SCOTT.EMP set ENAME='**PRIKEY**' where EMPNO=7369;

**LogMiner V$LOGMNR\_CONTENTS**SQL\_REDO  
--------------------------------------------------------------------------------  
update "SCOTT"."EMP" set "ENAME" = '**PRIKEY**' where "EMPNO" = '7369' and "ENAME" = 'SMITH' and ROWID = 'AAASZHAAEAAAACXAAA';

**GoldenGate LogDump**

2015 13:09:28.000.000 FieldComp            Len    28 RBA 1167

Name: SCOTT.EMP

After  Image:                                             Partition 4   G  s

 0000 000a 0000 0000 0000 0000 1cc9 0001 000a 0000 | ....................

 0006 5052 494b 4559                               | ..PRIKEY

Column     0 (x0000), Len    10 (x000a)

 0000 0000 0000 0000 1cc9                          | ..........

Column     1 (x0001), Len    10 (x000a)

 0000 0006 5052 494b 4559                          | ....**PRIKEY**

NOTE: The Primary Key value 7369 is now part of the Trail file record i.e. Column 0 EMPNO is HEX 1cc9 = DEC 7369.

***With All Columns***

**Oracle database command (and for GoldenGate version 11)  
alter table SCOTT.EMP add supplemental log data (all) columns;**

**GoldenGate version 12 TRANDATA equivalent**add trandata scott.emp allcols

**SQL executed**update SCOTT.EMP set ENAME='**ALLKEY**' where EMPNO=7566;

**LogMiner V$LOGMNR\_CONTENTS**  
SQL\_REDO  
--------------------------------------------------------------------------------  
update "SCOTT"."EMP" set "ENAME" = '**ALLKEY**' where "EMPNO" = '7566' and "ENAME" = 'JONES' and "JOB" = 'MANAGER' and "MGR" = '7839' and "HIREDATE" = TO\_DATE('02-APR-81', 'DD-MON-RR') and "SAL" = '2975' and "COMM" IS NULL and "DEPTNO" = '20' and ROWID = 'AAASZHAAEAAAACXAAD';

**GoldenGate LogDump**

2015/10/15 13:16:40.000.000 FieldComp            Len   124 RBA 1307

Name: SCOTT.EMP

After  Image:                                             Partition 4   G  s

 0000 000a 0000 0000 0000 0000 1d8e 0001 000a 0000 | ....................

 0006 414c 4c4b 4559 0002 000b 0000 0007 4d41 4e41 | ..ALLKEY........MANA

 4745 5200 0300 0a00 0000 0000 0000 001e 9f00 0400 | GER.................

 1500 0031 3938 312d 3034 2d30 323a 3030 3a30 303a | ...1981-04-02:00:00:

 3030 0005 000a 0000 0000 0000 0004 8a1c 0006 000a | 00..................

 ffff 0000 0000 0000 0000 0007 000a 0000 0000 0000 | ....................

 0000 0014                                         | ....

Column     0 (x0000), Len    10 (x000a)

 0000 0000 0000 0000 1d8e                          | ..........

Column     1 (x0001), Len    10 (x000a)

 0000 0006 414c 4c4b 4559                          | ....**ALLKEY**

Column     2 (x0002), Len    11 (x000b)

 0000 0007 4d41 4e41 4745 52                       | ....MANAGER

Column     3 (x0003), Len    10 (x000a)

 0000 0000 0000 0000 1e9f                          | ..........

Column     4 (x0004), Len    21 (x0015)

 0000 3139 3831 2d30 342d 3032 3a30 303a 3030 3a30 | ..1981-04-02:00:00:0

 30                                                | 0

Column     5 (x0005), Len    10 (x000a)

 0000 0000 0000 0004 8a1c                          | ..........

Column     6 (x0006), Len    10 (x000a)

 ffff 0000 0000 0000 0000                          | ..........

Column     7 (x0007), Len    10 (x000a)

 0000 0000 0000 0000 0014                          | ..........

NOTE: Now, all 8 columns of the SCOTT.EMP table and their values have been included in the Trail file, where Column 0 EMPNO is HEX 1d8e = DEC 7566.

**TEST CASE DETAILS**

The following test case details are provided to demonstrate how the results have been obtained.  
It is assumed that the user issuing the test case statements has the DBA Role i.e. connect / as SYSDBA.

***SCOTT.EMP create table statement***

This is the default SCOTT.EMP create statement, that includes the Primary Key on the EMPNO column, with no Supplemental Logging enabled.

SQL> select sys.dbms\_metadata.get\_ddl('TABLE','EMP','SCOTT') from dual;

SYS.DBMS\_METADATA.GET\_DDL('TABLE','EMP','SCOTT')

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

CREATE TABLE "SCOTT"."EMP"

 (    "EMPNO" NUMBER(4,0),

      "ENAME" VARCHAR2(10),

      "JOB" VARCHAR2(9),

      "MGR" NUMBER(4,0),

      "HIREDATE" DATE,

      "SAL" NUMBER(7,2),

      "COMM" NUMBER(7,2),

      "DEPTNO" NUMBER(2,0),

       CONSTRAINT "PK\_EMP" PRIMARY KEY ("EMPNO")

 USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

 STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

 TABLESPACE "USERS"  ENABLE,

       CONSTRAINT "FK\_DEPTNO" FOREIGN KEY ("DEPTNO")

        REFERENCES "SCOTT"."DEPT" ("DEPTNO") ENABLE

 ) SEGMENT CREATION IMMEDIATE PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

 STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

 TABLESPACE "USERS"

***SCOTT.EMP Rows***

The following rows are loaded into the SCOTT.EMP table when a database is created with the demonstration SCOTT schema.

SQL> select empno, ename, deptno from scott.emp;

     EMPNO ENAME          DEPTNO

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

      7369 SMITH              20

      7499 ALLEN              30

      7521 WARD               30

      7566 JONES              20

      7654 MARTIN             30

      7698 BLAKE              30

      7782 CLARK              10

      7788 SCOTT              20

      7839 KING               10

      7844 TURNER             30

      7876 ADAMS              20

      7900 JAMES              30

      7902 FORD               20

      7934 MILLER             10

***LogMiner commands***

The following SQL statements are to be used to load the new Archived log for mining.  
However, do not run them until indicated.

Copy the Archived log filename obtained from the V$ARCHIVED\_LOG query, and provide it when prompted by the ADD\_LOGFILE command.

The V$LOGMNR\_CONTENTS.SQL\_REDO column is queried to confirm what statements have actually been captured in the Redo/Archived log.

alter system switch logfile; /\* Switch the current Redo log, and in the process, an Archived log is created \*/

SELECT NAME FROM V$ARCHIVED\_LOG WHERE FIRST\_TIME = (SELECT MAX(FIRST\_TIME) FROM V$ARCHIVED\_LOG); /\* Identify the log \*/

EXECUTE DBMS\_LOGMNR.ADD\_LOGFILE(LOGFILENAME =>'&Name' , OPTIONS => DBMS\_LOGMNR.NEW); /\* Add the log for mining \*/

EXECUTE DBMS\_LOGMNR.START\_LOGMNR(OPTIONS => DBMS\_LOGMNR.DICT\_FROM\_ONLINE\_CATALOG); /\* Start the LogMiner session \*/

SELECT SQL\_REDO FROM V$LOGMNR\_CONTENTS WHERE SEG\_OWNER='SCOTT' and SEG\_NAME='EMP'; /\* Obtain the SQL statements that were issued \*/

EXECUTE DBMS\_LOGMNR.END\_LOGMNR(); /\* Stop the LogMiner session \*/

***Preparing the Oracle database***

To prepare the Oracle database for this test, we need to make sure that the database is actually in Archived log mode, and then switch to a new Redo log, so that there are no other SQL statements therein.  
Before the actual switch, query the Supplemental Logging DBA tables to confirm what is currently in effect.   
This is for comparison later on, when we add and change the Supplemental Logging.

connect / as sysdba

archive log list;

select SUPPLEMENTAL\_LOG\_DATA\_MIN, SUPPLEMENTAL\_LOG\_DATA\_PK, SUPPLEMENTAL\_LOG\_DATA\_UI, FORCE\_LOGGING from V$DATABASE;

SUPPLEME SUP SUP FOR

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

NO       NO  NO  NO

select LOG\_GROUP\_NAME, TABLE\_NAME, ALWAYS from DBA\_LOG\_GROUPS where OWNER='SCOTT';

no rows selected

select LOG\_GROUP\_NAME, COLUMN\_NAME, POSITION from DBA\_LOG\_GROUP\_COLUMNS where OWNER='SCOTT' and TABLE\_NAME = 'EMP';

no rows selected

alter system switch logfile;

The test is now ready to proceed.

***Test #1: Oracle database level Supplemental logging***

For this test, we need to first turn on Supplemental logging at the database level.  
Then, we'll update the SCOTT.EMP table, and examine the Redo generated.

**ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;**

/\* Check what Supplemental Logging is in effect \*/

select SUPPLEMENTAL\_LOG\_DATA\_MIN, SUPPLEMENTAL\_LOG\_DATA\_PK, SUPPLEMENTAL\_LOG\_DATA\_UI, FORCE\_LOGGING from V$DATABASE;

SUPPLEME SUP SUP FOR

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

YES      NO  NO  NO

select LOG\_GROUP\_NAME, TABLE\_NAME, ALWAYS from DBA\_LOG\_GROUPS where OWNER='SCOTT';

no rows selected

select LOG\_GROUP\_NAME, COLUMN\_NAME, POSITION from DBA\_LOG\_GROUP\_COLUMNS where OWNER='SCOTT' and TABLE\_NAME = 'EMP';

no rows selected

/\* Update the SCOTT.EMP table \*/

update SCOTT.EMP set ENAME='DBSUPLOG' where EMPNO=7934;

commit;

Mine the log (as per the [LogMiner commands](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=191858855383153&parent=EXTERNAL_SEARCH&sourceId=TROUBLESHOOTING&id=2067514.1&_afrWindowMode=0&_adf.ctrl-state=147fy3txw5_4#LogMinerCommands) above).  
The expected results are as follows;

SELECT SQL\_REDO FROM V$LOGMNR\_CONTENTS WHERE SEG\_OWNER='SCOTT' and SEG\_NAME='EMP';

SQL\_REDO

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

update "SCOTT"."EMP" set "ENAME" = 'DBSUPLOG' where "ENAME" = 'MILLER' and ROWID = 'AAASZHAAEAAAACXAAN';

Here we can see that the statement has been optimized to just the ENAME column, plus its' old and new values.  
This occurs because there is only 1 "MILLER" in a table that fits in 1 database block (unit of on-disk storage).

***Test #2: Primary Key column***

For this test, in addition to the database-level Supplemental Logging, we'll add Primary Key logging for the SCOTT.EMP table.  
Remember, the Primary Key for the EMP table is column EMPNO.

alter table SCOTT.EMP add supplemental log data (primary key) columns;

/\* Check what Supplemental Logging is in effect \*/

select SUPPLEMENTAL\_LOG\_DATA\_MIN, SUPPLEMENTAL\_LOG\_DATA\_PK, SUPPLEMENTAL\_LOG\_DATA\_UI, FORCE\_LOGGING from V$DATABASE;

SUPPLEME SUP SUP FOR

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

YES      NO  NO  NO

select LOG\_GROUP\_NAME, TABLE\_NAME, LOG\_GROUP\_TYPE, ALWAYS from DBA\_LOG\_GROUPS where OWNER='SCOTT';

LOG\_GROUP\_NAME                 TABLE\_NAME                     LOG\_GROUP\_TYPE               ALWAYS

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

SYS\_C0017767                   EMP                            PRIMARY KEY LOGGING ALWAYS

select LOG\_GROUP\_NAME, COLUMN\_NAME, POSITION from DBA\_LOG\_GROUP\_COLUMNS where OWNER='SCOTT' and TABLE\_NAME = 'EMP';

no rows selected

/\* Update the SCOTT.EMP table \*/

update SCOTT.EMP set ENAME='PRIKEY' where EMPNO=7369;

commit;

Here we see a system-generated Log Group for PRIMARY KEY LOGGING has been created, but no actual columns are (individually) logged.

Mine the log (as per the [LogMiner commands](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=191858855383153&parent=EXTERNAL_SEARCH&sourceId=TROUBLESHOOTING&id=2067514.1&_afrWindowMode=0&_adf.ctrl-state=147fy3txw5_4#LogMinerCommands) above).  
The expected results are as follows;

SELECT SQL\_REDO FROM V$LOGMNR\_CONTENTS WHERE SEG\_OWNER='SCOTT' and SEG\_NAME='EMP';

SQL\_REDO

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

alter table SCOTT.EMP add supplemental log data (primary key) columns;

update "SCOTT"."EMP" set "ENAME" = 'PRIKEY' where "EMPNO" = '7369' and "ENAME" = 'SMITH' and ROWID = 'AAASZHAAEAAAACXAAA';

This time, we see that the Primary Key column EMPNO and its' value of 7369 has been included in the SQL Redo statement.  
The actual ALTER TABLE statement is also captured, because it too is recorded with SEG\_OWNER='SCOTT' and SEG\_NAME='EMP'.  
Drop the Primary Key logging and switch to a new logfile.

alter table SCOTT.EMP drop supplemental log data (primary key) columns;

alter system switch logfile;

We are now ready for the next test.

***Test #3: All Columns***

Finally, we'll use the DATA (ALL) COLUMNS clause, to confirm its' effect on Supplemental Logging.

**alter table SCOTT.EMP add supplemental log data (all) columns;**

/\* Check what Supplemental Logging is in effect \*/

select SUPPLEMENTAL\_LOG\_DATA\_MIN, SUPPLEMENTAL\_LOG\_DATA\_PK, SUPPLEMENTAL\_LOG\_DATA\_UI, FORCE\_LOGGING from V$DATABASE;

SUPPLEME SUP SUP FOR

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

YES      NO  NO  NO

select LOG\_GROUP\_NAME, TABLE\_NAME, LOG\_GROUP\_TYPE, ALWAYS from DBA\_LOG\_GROUPS where OWNER='SCOTT';

LOG\_GROUP\_NAME                 TABLE\_NAME                     LOG\_GROUP\_TYPE               ALWAYS

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

SYS\_C0017769                   EMP                            ALL COLUMN LOGGING           ALWAYS

select LOG\_GROUP\_NAME, COLUMN\_NAME, POSITION from DBA\_LOG\_GROUP\_COLUMNS where OWNER='SCOTT' and TABLE\_NAME = 'EMP';

no rows selected

update SCOTT.EMP set ENAME='ALLKEY' where EMPNO=7566;

commit;

This time, we have a System generated Log Group for ALL COLUMN LOGGING.

Mine the log (as per the [LogMiner commands](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=191858855383153&parent=EXTERNAL_SEARCH&sourceId=TROUBLESHOOTING&id=2067514.1&_afrWindowMode=0&_adf.ctrl-state=147fy3txw5_4#LogMinerCommands) above).  
The expected results are as follows;

SQL\_REDO  
--------------------------------------------------------------------------------  
alter table SCOTT.EMP add supplemental log data (all) columns;  
update "SCOTT"."EMP" set "ENAME" = 'ALLKEY' where "EMPNO" = '7566' and "ENAME" = 'JONES' and "JOB" = 'MANAGER' and "MGR" = '7839' and "HIREDATE" = TO\_DATE('02-APR-81', 'DD-MON-RR') and "SAL" = '2975' and "COMM" IS NULL and "DEPTNO" = '20' and ROWID = 'AAASZHAAEAAAACXAAD';

Now, we see all 8 columns of the EMP table have been included in the Redo.  
This confirms that with the DATA (ALL) COLUMNS option, we don't get just all the keys, we get all the columns in the table, even though it is documented in the "...**key\_clause**".

Drop the All (column) logging and switch to a new logfile.

**alter table SCOTT.EMP drop supplemental log data (all) columns;**

**alter system switch logfile;**

We have now concluded our test.

***Restore the database***

Use the following commands to return the database and the SCOTT.EMP table back to its' previous settings/values.

ALTER DATABASE DROP SUPPLEMENTAL LOG DATA;

update SCOTT.EMP set ENAME='MILLER' where EMPNO=7934;

update SCOTT.EMP set ENAME='SMITH' where EMPNO=7369;

update SCOTT.EMP set ENAME='JONES' where EMPNO=7566;

commit;

**alter table SCOTT.EMP drop supplemental log data (all) columns;**

**etc**

***GoldenGate GGSCI commands***

The following are examples of the commands used to create a GoldenGate EXTRACT, for the purposes of creating a Trail file.  
It is assumed that the GGUSER has already been created, and granted the DBA role.

**EXTRACT parameters**EXTRACT <EXTNAME>  
EXTTRAIL ./dirdat/eo  
USERID <USERNAME, PASSWORD <PWD>  
TABLE SCOTT.\*;

**GGSCI commands**  
ADD EXTRACT <EXTNAME>, TRANLOG, BEGIN 2015-10-14:08:00 **\***  
ADD EXTTRAIL ./dirdat/eo, EXTRACT <EXTNAME>, MEGABYTES 50

**\*** Modify the date to when the Archived logs were actually created.

**LogMiner commands**  
open .\dirdat\eo000000  
ghdr on  
ggstoken detail  
detail on  
detail data  
next

NOTE: Use these options to get the most detail from the dump of the Trail record