

Oracle GoldenGate - Redo Processing

This page describes how GoldenGate processes redo generated by an Oracle database. It is based on original research and therefore cannot be guaranteed to be correct.

The research was performed in Oracle 11.2.0.3 on OEL 5.6 running in Oracle VirtualBox. The Oracle GoldenGate version was 11.2.1.0.1

This page describes the trails created using the Classic Capture method.

Introduction

Transactions

GoldenGate appears to extract data when a transaction commits. It uses the additional information stored in the redo log to identify the Oracle transaction ID, which also identifies the transaction in the GoldenGate trail.

If a transaction is rolled back, then it is not stored in the GoldenGate trail at all.

In the case of long running transactions, not all of the changes may still be available in the online redo log when the transaction is committed. In this case it is necessary for GoldenGate to inspect the archived redo logs for some of the earlier changes. Therefore the archived redo logs must still be available. This may affect sites which back up and delete their archived redo logs on a regular basis e.g. every two hours.

Further research is required to verify the behaviour of GoldenGate in the case of long running transactions.

Each transaction is identified by a transaction ID. The transaction ID consists of three components:

- Undo segment number
- Undo slot number
- Sequence number

The undo segment is allocated when a session first issues a DML change. Undo segments are not allocated for SELECT-only statements.

Each undo segment contains an undo header block. This block contains an array of slots. A slot must be reserved by a transaction to indicate that it is in progress. When the transaction completes the slot header is updated again. If the session fails for any reason, then the state of the undo header is used to determine whether any rollback is required for the failed transaction.

The transaction ID can be determined from the V\$TRANSACTION dynamic performance view. For example:

```
SQL> SELECT xidusn, xidslot, xidsqn FROM v$transaction;
```

XIDUSN	XIDSLOT	XIDSQN
6	28	1186

The transaction ID is also recorded within the online redo log. For opcode 5.2 which reserves a slot in the undo header for a new transaction, it is stored in the ktudh structure. For example:

```
CHANGE #1 TYP:0 CLS:27 AFN:3 DBA:0x00c000d0 OBJ:4294967295 SCN:0x0000.0017f125  
SEQ:1 OP:5.2 ENC:0 RBL:0  
ktudh redo: slt: 0x001c sqn: 0x000004a2 flg: 0x0012 siz: 136 fbi: 0  
uba: 0x00c01b31.0115.07 pxid: 0x0000.000.00000000
```

In the above example, the slot number (slt) is 0x1c (28) and the sequence number (sqn) is 0x4a2 (1186). The undo segment number can be derived from the class (CLS). Class 27 is the undo header for undo segment number 6.

```

CHANGE #2 TYP:0 CLS:28 AFN:3 DBA:0x00c01b31 OBJ:4294967295 SCN:0x0000.0017f124
SEQ:1 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 136 spc: 7310 flg: 0x0012 seq: 0x0115 rec: 0x07
          xid: 0x0006.01c.000004a2
ktubl redo: slt: 28 rci: 0 opc: 11.1 [objn: 75577 objd: 75577 tsn: 4]
Undo type: Regular undo          Begin trans          Last buffer split: No
Temp Object: No
Tablespace Undo: No
          0x00000000 prev ctl uba: 0x00c01b31.0115.06
prev ctl max cmt scn: 0x0000.0017e8b0 prev tx cmt scn: 0x0000.0017e8c4
txn start scn: 0xffff.ffffffff logon user: 84 prev brb: 12589869 prev bcl: 0
BuExt idx: 0 flg2: 0
KDO undo record:
KTB Redo
op: 0x03 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: Z
KDO Op code: DRP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x0100010c hdba: 0x010000aa
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 87(0x57)

```

For opcode 5.1 which creates an undo record for an existing transaction, the transaction ID (xid) is stored in the ktudb structure. In the above example it is reported as 0x0006.01c.000004a2.

```

CHANGE #3 TYP:2 CLS:1 AFN:4 DBA:0x0100010c OBJ:75577 SCN:0x0000.0017cd99 SEQ:1
OP:11.2 ENC:0 RBL:0
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0006.01c.000004a2 uba: 0x00c01b31.0115.07
KDO Op code: IRP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x0100010c hdba: 0x010000aa
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 87(0x57) size/delt: 27
fb: --H-FL-- lb: 0x2 cc: 4
null: ----
col 0: [ 2] c3 0b
col 1: [ 6] 4f 57 4e 45 52 31
col 2: [ 7] 4f 42 4a 45 43 54 31
col 3: [ 5] 54 41 42 4c 45

```

For opcode 11.2 which inserts a row piece (IRP), the transaction ID is again reported as xid (0x0006.01c.000004a2)

```

CHANGE #4 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:5.20 ENC:0
session number = 50
serial number = 99
transaction name =
version 186647296
audit sessionid 210055
Client Id =
login username = US01

```

The transaction ID is not included in the opcode 5.20 structure that is written to the redo log once for each transaction. This structure includes additional information that is required by tools such as LogMiner to recreate transactions. As GoldenGate has similar requirements, it may also use this information.

The transaction ID is recorded in the GoldenGate extract trail for each change in the transaction. For example (TrailAnalyzer output):

```

<47_54_36>
# Transaction ID
6.28.1186

```

Undo

GoldenGate only captures committed transactions. Therefore it never needs to capture undo headers or undo records from the online redo log. This significantly reduces the size of the trail file in comparison with the online redo log.

Undo will be generated in the target database when changes are applied by the replication process.

Although undo changes are not extracted or replicated directly, each transaction replicated by GoldenGate will need to reserve a slot in an undo segment header and then to generate undo records in the undo segments for each row updated by the transaction. There is relationship between the undo segments used in the source and target database for a specific transaction.

Block Cleanouts

When a transaction updates a block, it reserves a slot in the transaction header at the top of the block. It then places a lock in the row header of each row that it updates on the block. These locks are retained until the transaction commits or rolls back. The block remains as a dirty block in the buffer cache initially, but may subsequently be written back to its data file if additional space is required in the buffer cache.

When the transaction commits, the row locks must be cleared and the slot in the transaction header must be released. However, this may not be efficient for all transactions. For long-running transactions, many blocks may have been written back to disk, and for these the block cleanout mechanism is used.

When a block is read, the transaction headers are inspected. If any slot has been reserved it is compared with the undo header to determine if it has subsequently committed. If the transaction has committed, then all row locks for that transaction are unset and the slot in the transaction header is released. The block is then marked as dirty so that it can be written back to disk again asynchronously.

Block cleanouts are unusual in that they occur in SELECT statements. Thus a SQL statement that appears to be a read-only operation can initiate significant amounts of write I/O. This I/O can be difficult to reproduce as it will only occur the first time a block with committed transactions is discovered. Therefore running the same SELECT statement a second time may visit the same set of blocks, but may not require any further block cleanouts.

Redo is generated for each block cleanout operation. The redo operation code is 4.1. The block cleanout operation is recursive and always occupies a separate redo record. The redo record is at least 72 bytes of which the redo record header is 36 bytes and the block cleanout change is at least 36 bytes. The size of the block cleanout change is dependent on the number of committed transactions that are cleared out in the block.

GoldenGate ignores block cleanout operations in the online redo logs. They are not extracted and consequently will not be replicated. Block cleanouts may still occur in the replication database, especially if replicated blocks from long-running transactions are subsequently queried, for example in a reporting database.

Indexes

DML changes to tables generate level 11 redo operations. If indexes are affected by the change, then level 10 redo is generated for each redo change.

Generally index changes will consist of an undo change and a redo change. An INSERT operation will require an index addition for each affected index; a DELETE operation will require an index deletion for each affected index. An UPDATE operation requires an index deletion followed by an index addition for each affected index. There is no index update redo operation.

With the exception of Index-Organized Tables (IOTs), GoldenGate does not replicate index changes. This significantly reduced the amount of data extracted from the online redo log. However, each change replicated by GoldenGate will maintain any indexes created against the table in the target database.

It is not necessary for the source and target tables to have identical index definitions. For example if the target database is a data warehouse, new indexes may be required to support the workload. Similarly indexes required to optimize an OLTP workload in the source database may not be required in the target database.

Examples

This section contains examples of how DML changes are represented in Oracle Redo Logs and GoldenGate Trails.

The examples are based on the following objects:

```
CREATE TABLE course
(
  university VARCHAR2(30),
  subject    VARCHAR2(30),
  entry_year NUMBER
);

ALTER TABLE course
ADD CONSTRAINT course_pk
PRIMARY KEY (university,subject,entry_year);

CREATE TABLE student
(
  student_key NUMBER,
  first_name  VARCHAR2(30),
  surname     VARCHAR2(30),
  gender      VARCHAR2(1),
  university  VARCHAR2(30),
  subject     VARCHAR2(30),
  entry_year  NUMBER,
  tuition_fee NUMBER
);

ALTER TABLE student
ADD CONSTRAINT student_pk
PRIMARY KEY (student_key);

ALTER TABLE student
ADD CONSTRAINT student_uk
UNIQUE (first_name,surname);

ALTER TABLE student
ADD CONSTRAINT student_course
FOREIGN KEY (university,subject,entry_year)
REFERENCES course (university,subject,entry_year);
```

The COURSE table is populated with the following initial data:

```
INSERT INTO course VALUES ('Oxford','Theology',2013);
INSERT INTO course VALUES ('Oxford','Biology',2013);
INSERT INTO course VALUES ('Oxford','Biology',2014);
INSERT INTO course VALUES ('Cambridge','Economics',2013);
INSERT INTO course VALUES ('Cambridge','Physics',2013);
INSERT INTO course VALUES ('Cambridge','Chemistry',2013);
INSERT INTO course VALUES ('Manchester','Chemistry',2013);
INSERT INTO course VALUES ('Manchester','Computer Science',2013);
INSERT INTO course VALUES ('Manchester','Art History',2013);
INSERT INTO course VALUES ('Manchester','Media Studies',2013);

COMMIT;
```

The STUDENT table is populated with the following initial data:

```
INSERT INTO student VALUES (1001,'Lucy','Brotherton','F',
'Cambridge','Chemistry',2013,9000);

INSERT INTO student VALUES (1002,'Rebecca','Brown','F',
'Oxford','Biology',2013,9000);
```

```

INSERT INTO student VALUES (1003,'Simon','Campbell','M',
'Cambridge','Physics',2013,7500);

INSERT INTO student VALUES (1004,'Jason','Robinson','M',
'Oxford','Biology',2013,7500);

INSERT INTO student VALUES (1005,'Stuart','Overy','M',
'Manchester','Art History',2013,9000);

INSERT INTO student VALUES (1006,'Tom','Homer','M',
'Manchester','Computer Science',2013,9000);

INSERT INTO student VALUES (1007,'Victoria','Evans','F',
'Oxford','Theology',2013,9000);

INSERT INTO student VALUES (1008,'Katy','Pierce','F',
'Oxford','Theology',2013,9000);

INSERT INTO student VALUES (1009,'Shane','Thomas','M',
'Manchester','Media Studies',2013,9000);

INSERT INTO student VALUES (1010,'Sarah','McCloud','F',
'Oxford','Biology',2014,9000);

```

The tables are not fully normalized. This is deliberate policy to make the redo log dumps and the GoldenGate trails easier to interpret. I have used alphanumeric VARCHAR2 data of varying lengths where possible. I acknowledge that this is not good database design, but it is intended for illustrative purposes only.

Single Row Insertion

Consider the following statements:

```

INSERT INTO student VALUES (1011,'Jordan','Sherwood','M',
'Manchester','Chemistry',2013,9000);

COMMIT;

```

The above statement generates the following entries in the redo log. The first redo record inserts the new row into the STUDENT table:

```

REDO RECORD - Thread:1 RBA: 0x000044.00000005.0010 LEN: 0x0228 VLD: 0x05
SCN: 0x0000.0018bcde SUBSCN: 1 03/31/2013 23:59:58
(LWN RBA: 0x000044.00000005.0010 LEN: 0003 NST: 0001 SCN: 0x0000.0018bcde)

CHANGE #1 TYP:0 CLS:23 AFN:3 DBA:0x00c000b0 OBJ:4294967295 SCN:0x0000.0018bcb0
SEQ:1 OP:5.2 ENC:0 RBL:0
ktudh redo: slt: 0x000b sqn: 0x00000356 flg: 0x0012 siz: 136 fbi: 0
          uba: 0x00c000b2.01ee.0d      pxid: 0x0000.000.00000000

CHANGE #2 TYP:0 CLS:24 AFN:3 DBA:0x00c000b2 OBJ:4294967295 SCN:0x0000.0018bcaf
SEQ:2 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 136 spc: 6418 flg: 0x0012 seq: 0x01ee rec: 0x0d
          xid: 0x0004.00b.00000356
ktubl redo: slt: 11 rci: 0 opc: 11.1 [objn: 76490 objd: 76495 tsn: 4]
Undo type: Regular undo          Begin trans      Last buffer split: No
Temp Object: No
Tablespace Undo: No
          0x00000000 prev ctl uba: 0x00c000b2.01ee.0b
prev ctl max cmt scn: 0x0000.0018afaa prev tx cmt scn: 0x0000.0018afdb
txn start scn: 0xffff.ffffffff logon user: 90 prev brb: 12584964 prev bcl: 0
BuExt idx: 0 flg2: 0
KDO undo record:
KTB Redo
op: 0x03 ver: 0x01
compat bit: 4 (post-11) padding: 1

```

```

op: Z
KDO Op code: DRP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 10(0xa)

CHANGE #3 TYP:2 CLS:1 AFN:4 DBA:0x01000436 OBJ:76495 SCN:0x0000.00187e82 SEQ:1
OP:11.2 ENC:0 RBL:0
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0004.00b.00000356 uba: 0x00c000b2.01ee.0d
KDO Op code: IRP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 10(0xa) size/delt: 53
fb: --H-FL-- lb: 0x2 cc: 8
null: -----
col 0: [ 3] c2 0b 0c
col 1: [ 6] 4a 6f 72 64 61 6e
col 2: [ 8] 53 68 65 72 77 6f 6f 64
col 3: [ 1] 4d
col 4: [10] 4d 61 6e 63 68 65 73 74 65 72
col 5: [ 9] 43 68 65 6d 69 73 74 72 79
col 6: [ 3] c2 15 0e
col 7: [ 2] c2 5b

CHANGE #4 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:5.20 ENC:0
session number = 44
serial number = 325
transaction name =
version 186647296
audit sessionid 220157
Client Id =
login username = US03

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	68
1	5.2	Allocate transaction slot in undo header	60
2	5.1	Undo for insert operation (DRP)	180
3	11.2	Redo for insert operation (IRP)	180
4	5.20	Media recovery marker	64

The table has a primary key index. Therefore a second redo record is generated to add the new key to the index:

```

REDO RECORD - Thread:1 RBA: 0x000044.00000006.0090 LEN: 0x00f0 VLD: 0x01
SCN: 0x0000.0018bcde SUBSCN: 1 03/31/2013 23:59:58

CHANGE #1 TYP:0 CLS:24 AFN:3 DBA:0x00c000b2 OBJ:4294967295 SCN:0x0000.0018bcde
SEQ:1 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 92 spc: 6280 flg: 0x0022 seq: 0x01ee rec: 0x0e
  xid: 0x0004.00b.00000356
ktubu redo: slt: 11 rci: 13 opc: 10.22 objn: 76491 objd: 76494 tsn: 4
Undo type: Regular undo Undo type: Last buffer split: No
Tablespace Undo: No
  0x00000000
index undo for leaf key operations
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0009.001.00000409 uba: 0x00c00102.01b6.10
  flg: C--- lkc: 0 scn: 0x0000.0018bc6e
Dump kdilk : itl=2, kdxlklflg=0x1 sdc=0 indexid=0x100043a block=0x0100043b

```

```

(kdxlpu): purge leaf row
key : (4): 03 c2 0b 0c

CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x0100043b OBJ:76494 SCN:0x0000.0018bcde SEQ:1
OP:10.2 ENC:0 RBL:0
index redo (kdxlin): insert leaf row
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0004.00b.00000356 uba: 0x00c000b2.01ee.0e
REDO: SINGLE / -- / --
itl: 2, sno: 10, row size 14
insert key: (4): 03 c2 0b 0c
keydata: (6): 01 00 04 36 00 0a

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.1	Undo for insert leaf row operation	136
2	10.2	Redo for insert leaf row operation	80

The table also has a unique key index. Therefore a third redo record is generated to add the new key to the index:

```

REDO RECORD - Thread:1 RBA: 0x000044.00000006.01c8 LEN: 0x0108 VLD: 0x01
SCN: 0x0000.0018bcde SUBSCN: 1 03/31/2013 23:59:58

CHANGE #1 TYP:0 CLS:24 AFN:3 DBA:0x00c000b2 OBJ:4294967295 SCN:0x0000.0018bcde
SEQ:2 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 104 spc: 6186 flg: 0x0022 seq: 0x01ee rec: 0x0f
xid: 0x0004.00b.00000356
ktubu redo: slt: 11 rci: 14 opc: 10.22 objn: 76492 objd: 76493 tsn: 4
Undo type: Regular undo Undo type: Last buffer split: No
Tablespace Undo: No
0x00000000
index undo for leaf key operations
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0009.001.00000409 uba: 0x00c00102.01b6.11
flg: C--- lkc: 0 scn: 0x0000.0018bc6e
Dump kdilk : itl=2, kdxlkflg=0x1 sdc=0 indexid=0x1000342 block=0x01000343
(kdxlpu): purge leaf row
key : (16): 06 4a 6f 72 64 61 6e 08 53 68 65 72 77 6f 6f 64

CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x01000343 OBJ:76493 SCN:0x0000.0018bcde SEQ:1
OP:10.2 ENC:0 RBL:0
index redo (kdxlin): insert leaf row
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0004.00b.00000356 uba: 0x00c000b2.01ee.0f
REDO: SINGLE / -- / --
itl: 2, sno: 1, row size 26
insert key: (16): 06 4a 6f 72 64 61 6e 08 53 68 65 72 77 6f 6f 64
keydata: (6): 01 00 04 36 00 0a

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.1	Undo for insert leaf row operation	148
2	10.2	Redo for insert leaf row operation	92

Finally the COMMIT statement generates the following redo:

```
REDO RECORD - Thread:1 RBA: 0x000044.00000007.00e0 LEN: 0x00a4 VLD: 0x01
SCN: 0x0000.0018bcdF SUBSCN: 1 03/31/2013 23:59:58
CHANGE #1 TYP:0 CLS:23 AFN:3 DBA:0x00c000b0 OBJ:4294967295 SCN:0x0000.0018bcde
SEQ:1 OP:5.4 ENC:0 RBL:0
ktucm redo: slt: 0x000b sqn: 0x00000356 srt: 0 sta: 9 flg: 0x2 ktucf redo: uba:
0x00c000b2.01ee.0f ext: 0 spc: 6080 fbi: 0
CHANGE #2 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:24.4 ENC:0
```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.4	Release transaction slot in undo header	72
2	24.4	Media recovery marker	68

The redo log entries generate the following TrailAnalyzer output from the GoldenGate extract trail:

```
# Header: Type=47 (G) Flag=1 Len=224 (4 bytes)
# Row (216 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
# Row Header (47 bytes)
# Flags - 0x05 Insert
# Timestamp 2013:03:31 23:59:58 000000 ms
# Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=110 (4 bytes)
<47_44>
# Row Data (110 bytes)
# Slot 0 (8 bytes)
00 00 00 04 31 30 31 31 # STUDENT_KEY = 1011
# Slot 1 (10 bytes)
00 00 00 06 4A 6F 72 64 61 6E # FIRST_NAME = Jordan
# Slot 2 (12 bytes)
00 00 00 08 53 68 65 72 77 6F 6F 64 # SURNAME = Sherwood
# Slot 3 (5 bytes)
00 00 00 01 4D # GENDER = M
# Slot 4 (14 bytes)
00 00 00 0A 4D 61 6E 63 68 65 73 74 65 72 # UNIVERSITY = Manchester
# Slot 5 (13 bytes)
00 00 00 09 43 68 65 6D 69 73 74 72 79 # SUBJECT = Chemistry
# Slot 6 (8 bytes)
00 00 00 04 32 30 31 33 # ENTRY_YEAR = 2013
# Slot 7 (8 bytes)
00 00 00 04 39 30 30 30 # TUITION_FEE = 9000

# Header: Type=54 (T) Flag=0 Len=47 (4 bytes)
<47_54>
# Row Metadata (47 bytes)
# Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
<47_54_52>
# ROWID: 20 bytes
AAASrPAAEAAAAQ2AAK 1

# Header: Type=4C (L) Flag=0 Len=7 (4 bytes)
<47_54_4C>
# Commit SCN (7 bytes)
1621215

# Header: Type=36 (6) Flag=0 Len=8 (4 bytes)
<47_54_36>
# Transaction ID (8 bytes)
```


4.11.854

```
# Header: Type=5A (Z) Flag=1 Len=224 (4 bytes)
```

The index redo is not extracted into the trail.

The commit redo is not extracted explicitly into the trail.

The total number of bytes required in the GoldenGate trail for the above change is 225.

Single Row Updates

Consider the following statements:

```
UPDATE student SET tuition_fee = 6000
WHERE student_key = 1010;

COMMIT;
```

The above statement generates the following entries in the redo log. The first redo record updates an existing row into the STUDENT table:

```
REDO RECORD - Thread:1 RBA: 0x000045.00000006.0010 LEN: 0x024c VLD: 0x05
SCN: 0x0000.0018c373 SUBSCN: 1 04/01/2013 00:55:00
(LWN RBA: 0x000045.00000006.0010 LEN: 0002 NST: 0001 SCN: 0x0000.0018c373)

CHANGE #1 TYP:0 CLS:21 AFN:3 DBA:0x00c000a0 OBJ:4294967295 SCN:0x0000.0018c34c
SEQ:1 OP:5.2 ENC:0 RBL:0
ktudh redo: slt: 0x0006 sqn: 0x000003f4 flg: 0x0012 siz: 200 fbi: 0
           uba: 0x00c000a5.0156.1d pxid: 0x0000.000.00000000

CHANGE #2 TYP:0 CLS:22 AFN:3 DBA:0x00c000a5 OBJ:4294967295 SCN:0x0000.0018c34b
SEQ:2 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 200 spc: 2246 flg: 0x0012 seq: 0x0156 rec: 0x1d
           xid: 0x0003.006.000003f4
ktubl redo: slt: 6 rci: 0 opc: 11.1 [objn: 76490 objd: 76495 tsn: 4]
Undo type: Regular undo          Begin trans      Last buffer split: No
Temp Object: No
Tablespace Undo: No
           0x00000000 prev ctl uba: 0x00c000a5.0156.1b
prev ctl max cmt scn: 0x0000.0018b7ee prev tx cmt scn: 0x0000.0018b7fd
txn start scn: 0xffff.ffffffff logon user: 90 prev brb: 12583223 prev bcl: 0
BuExt idx: 0 flg2: 0
KDO undo record:
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x000a.014.00000345 uba: 0x00c007ff.00ec.1c
           flg: C--- lkc: 0 scn: 0x0000.00187e82
KDO Op code: URP row dependencies Disabled
xtype: XA flags: 0x00000000bdba: 0x01000436 hdba: 0x01000432
itli: 1 ispac: 0 maxfr: 4858
tabn: 0 slot: 9(0x9) flag: 0x2c lock: 0 ckix: 0
ncol: 8 nnew: 1 size: 0
col 7: [ 2] c2 5b

CHANGE #3 TYP:2 CLS:1 AFN:4 DBA:0x01000436 OBJ:76495 SCN:0x0000.0018bcdf SEQ:1
OP:11.5 ENC:0 RBL:0
KTB Redo
op: 0x11 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0003.006.000003f4 uba: 0x00c000a5.0156.1d
Block cleanout record, scn: 0x0000.0018c373 ver: 0x01 opt: 0x02, entries follow...
itli: 1 flg: 2 scn: 0x0000.00187e82
itli: 2 flg: 2 scn: 0x0000.0018bcdf
```

```

KDO Op code: URP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 1 ispac: 0 maxfr: 4858
tabn: 0 slot: 9(0x9) flag: 0x2c lock: 1 ckix: 0
ncol: 8 nnew: 1 size: 0
col 7: [ 2] c2 3d

CHANGE #4 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:5.20 ENC:0
session number = 56
serial number = 101
transaction name =
version 186647296
audit sessionid 220171
Client Id =
login username = US03

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	68
1	5.2	Allocate transaction slot in undo header	60
2	5.1	Undo for update operation (URP)	248
3	11.5	Redo for update operation (URP)	148
4	5.20	Media recovery marker	64

In this example, no indexes are updated as the TUITION_FEE column is not included in any indexes.

The COMMIT statement generates the following redo:

```

REDO RECORD - Thread:1 RBA: 0x000045.00000007.006c LEN: 0x00a4 VLD: 0x01
SCN: 0x0000.0018c374 SUBSCN: 1 04/01/2013 00:55:00
CHANGE #1 TYP:0 CLS:21 AFN:3 DBA:0x00c000a0 OBJ:4294967295 SCN:0x0000.0018c373
SEQ:1 OP:5.4 ENC:0 RBL:0
ktucm redo: slt: 0x0006 sqn: 0x000003f4 srt: 0 sta: 9 flg: 0x2 ktucf redo: uba:
0x00c000a5.0156.1d ext: 0 spc: 2044 fbi: 0
CHANGE #2 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:24.4 ENC:0

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.4	Release transaction slot in undo header	72
2	24.4	Media recovery marker	68

The redo log entries generate the following TrailAnalyzer output from the GoldenGate extract trail:

```

# Header: Type=47 (G) Flag=1 Len=138 (4 bytes)

# Row (130 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
  # Row Header (47 bytes)
  # Flags - 0x0F Update
  # Timestamp 2013:04:01 00:55:00 000000 ms
  # Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=24 (4 bytes)
<47_44>
  # Row Data (24 bytes)
  # Slot 0 (8 bytes)
  00 00 00 04 31 30 31 30
  # Slot 7 (8 bytes)
  # STUDENT_KEY = 1010

```

```

00 00 00 04 36 30 30 30 # TUITION_FEE = 6000

# Header: Type=54 (T) Flag=0 Len=47 (4 bytes)
<47_54>
# Row Metadata (47 bytes)
# Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
<47_54_52>
# ROWID: 20 bytes
AAASrPAAEAAAAQ2AAJ 1

# Header: Type=4C (L) Flag=0 Len=7 (4 bytes)
<47_54_4C>
# Commit SCN (7 bytes)
1622900

# Header: Type=36 (6) Flag=0 Len=8 (4 bytes)
<47_54_36>
# Transaction ID (8 bytes)
3.6.1012

# Header: Type=5A (Z) Flag=1 Len=138 (4 bytes)

```

The index redo is not extracted into the trail.

The commit redo is not extracted explicitly into the trail.

The total number of bytes required in the GoldenGate trail for the above change is 138.

Note that the primary key is included in the trail. This is required to identify the row in the target database. Although the ROWID is included in the trail, it cannot be used to locate the row in the target database as the ROWIDs can differ between the source and target databases.

The primary key value is not explicitly stored in the redo change vector; it is included in the supplemental data for the undo change vector.

Single row deletes

Consider the following statements:

```

DELETE FROM student WHERE student_key = 1004;

COMMIT;

```

The above statement generates the following entries in the redo log. The first redo record deletes an existing row into the STUDENT table:

```

REDO RECORD - Thread:1 RBA: 0x000046.00000006.0010 LEN: 0x0238 VLD: 0x05
SCN: 0x0000.0018cf24 SUBSCN: 1 04/01/2013 02:35:47
(LWN RBA: 0x000046.00000006.0010 LEN: 0003 NST: 0001 SCN: 0x0000.0018cf24)

CHANGE #1 TYP:0 CLS:17 AFN:3 DBA:0x00c00080 OBJ:4294967295 SCN:0x0000.0018ceff
SEQ:1 OP:5.2 ENC:0 RBL:0
ktudh redo: slt: 0x0021 sqn: 0x0000033e flg: 0x0012 siz: 256 fbi: 0
           uba: 0x00c00157.01da.05      pxid: 0x0000.000.00000000

CHANGE #2 TYP:0 CLS:18 AFN:3 DBA:0x00c00157 OBJ:4294967295 SCN:0x0000.0018cefe
SEQ:2 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 256 spc: 7506 flg: 0x0012 seq: 0x01da rec: 0x05
           xid: 0x0001.021.0000033e

ktubl redo: slt: 33 rci: 0 opc: 11.1 [objn: 76490 objd: 76495 tsn: 4]
Undo type: Regular undo      Begin trans      Last buffer split: No
Temp Object: No
Tablespace Undo: No
           0x00000000 prev ctl uba: 0x00c00157.01da.03

```

```

prev ctl max cmt scn: 0x0000.0018c1ec prev tx cmt scn: 0x0000.0018c329
txn start scn: 0xffff.ffffffff logon user: 90 prev brb: 12583253 prev bcl: 0
BuExt idx: 0 flg2: 0
KDO undo record:
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0004.00b.00000356 uba: 0x00c000b2.01ee.0d
      flg: C--- lkc: 0 scn: 0x0000.0018bcdcf
KDO Op code: IRP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 3(0x3) size/delt: 46
fb: --H-FL-- lb: 0x0 cc: 8
null: -----
col 0: [ 3] c2 0b 05
col 1: [ 5] 4a 61 73 6f 6e
col 2: [ 8] 52 6f 62 69 6e 73 6f 6e
col 3: [ 1] 4d
col 4: [ 6] 4f 78 66 6f 72 64
col 5: [ 7] 42 69 6f 6c 6f 67 79
col 6: [ 3] c2 15 0e
col 7: [ 2] c2 4c

CHANGE #3 TYP:2 CLS:1 AFN:4 DBA:0x01000436 OBJ:76495 SCN:0x0000.0018c374 SEQ:1
OP:11.3 ENC:0 RBL:0
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0001.021.0000033e uba: 0x00c00157.01da.05
KDO Op code: DRP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 3(0x3)

CHANGE #4 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:5.20 ENC:0
session number = 44
serial number = 531
transaction name =
version 186647296
audit sessionid 220192
Client Id =
login username = US03

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	68
1	5.2	Allocate transaction slot in undo header	60
2	5.1	Undo for delete operation (IRP)	300
3	11.3	Redo for delete operation (DRP)	76
4	5.20	Media recovery marker	64

The second redo record deletes the key from the primary key index:

```

REDO RECORD - Thread:1 RBA: 0x000046.00000007.00a0 LEN: 0x00ec VLD: 0x01
SCN: 0x0000.0018cf24 SUBSCN: 1 04/01/2013 02:35:47

CHANGE #1 TYP:0 CLS:18 AFN:3 DBA:0x00c00157 OBJ:4294967295 SCN:0x0000.0018cf24
SEQ:1 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 100 spc: 7248 flg: 0x0022 seq: 0x01da rec: 0x06
      xid: 0x0001.021.0000033e
ktubu redo: slt: 33 rci: 5 opc: 10.22 objn: 76491 objd: 76494 tsn: 4
Undo type: Regular undo Undo type: Last buffer split: No
Tablespace Undo: No

```

```

0x00000000
index undo for leaf key operations
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0004.00b.00000356 uba: 0x00c000b2.01ee.0e
           flg: C--- lkc: 0      scn: 0x0000.0018bcdf
Dump kdilk : itl=2, kdxlkflg=0x1 sdc=0 indexid=0x100043a block=0x0100043b
(kdxlre): restore leaf row (clear leaf delete flags)
key : (4): 03 c2 0b 05
keydata/bitmap: (6): 01 00 04 36 00 03

CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x0100043b OBJ:76494 SCN:0x0000.0018cf24 SEQ:1
OP:10.4 ENC:0 RBL:0
index redo (kdxlde): delete leaf row
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0001.021.0000033e uba: 0x00c00157.01da.06
REDO: SINGLE / -- / --
itl: 2, sno: 3, row size 14

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.1	Undo for insert leaf row operation	148
2	10.2	Redo for insert leaf row operation	64

The third redo record deletes the key from the unique key index:

```

REDO RECORD - Thread:1 RBA: 0x000046.00000007.01d4 LEN: 0x00f8 VLD: 0x01
SCN: 0x0000.0018cf24 SUBSCN: 1 04/01/2013 02:35:47

CHANGE #1 TYP:0 CLS:18 AFN:3 DBA:0x00c00157 OBJ:4294967295 SCN:0x0000.0018cf24
SEQ:2 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 112 spc: 7146 flg: 0x0022 seq: 0x01da rec: 0x07
           xid: 0x0001.021.0000033e
ktubu redo: slt: 33 rci: 6 opc: 10.22 objn: 76492 objd: 76493 tsn: 4
Undo type: Regular undo Undo type: Last buffer split: No
Tablespace Undo: No
0x00000000
index undo for leaf key operations
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0004.00b.00000356 uba: 0x00c000b2.01ee.0f
           flg: C--- lkc: 0      scn: 0x0000.0018bcdf
Dump kdilk : itl=2, kdxlkflg=0x1 sdc=0 indexid=0x1000342 block=0x01000343
(kdxlre): restore leaf row (clear leaf delete flags)
key : (15): 05 4a 61 73 6f 6e 08 52 6f 62 69 6e 73 6f 6e
keydata/bitmap: (6): 01 00 04 36 00 03

CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x01000343 OBJ:76493 SCN:0x0000.0018cf24 SEQ:1
OP:10.4 ENC:0 RBL:0
index redo (kdxlde): delete leaf row
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0001.021.0000033e uba: 0x00c00157.01da.07
REDO: SINGLE / -- / --
itl: 2, sno: 0, row size 25

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.1	Undo for insert leaf row operation	160
2	10.2	Redo for insert leaf row operation	64

The COMMIT statement generates the following redo:

```

REDO RECORD - Thread:1 RBA: 0x000046.00000008.00dc LEN: 0x00c0 VLD: 0x01
SCN: 0x0000.0018cf25 SUBSCN: 1 04/01/2013 02:35:47
CHANGE #1 TYP:0 CLS:17 AFN:3 DBA:0x00c00080 OBJ:4294967295 SCN:0x0000.0018cf24
SEQ:1 OP:5.4 ENC:0 RBL:0
ktucm redo: slt: 0x0021 sqn: 0x0000033e srt: 0 sta: 9 flg: 0x12 ktucf redo: uba:
0x00c00157.01da.07 ext: 3 spc: 7032 fbi: 0
CHANGE #2 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:24.4 ENC:0

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.4	Release transaction slot in undo header	100
2	24.4	Media recovery marker	68

The redo log entries generate the following TrailAnalyzer output from the GoldenGate extract trail:

```

# Header: Type=47 (G) Flag=1 Len=126 (4 bytes)

# Row (118 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
  # Row Header (47 bytes)
  # Flags - 0x03 Delete
  # Timestamp 2013:04:01 02:35:47 000000 ms
  # Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=12 (4 bytes)
<47_44>
  # Row Data (12 bytes)
  # Slot 0 (8 bytes)
  00 00 00 04 31 30 30 34          # STUDENT_KEY = 1004

# Header: Type=54 (T) Flag=0 Len=47 (4 bytes)
<47_54>
  # Row Metadata (47 bytes)
  # Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
  <47_54_52>
  # ROWID: 20 bytes
  AAASrPAAEAAAAQ2AAD 1

# Header: Type=4C (L) Flag=0 Len=7 (4 bytes)
<47_54_4C>
  # Commit SCN (7 bytes)
  1625893

# Header: Type=36 (6) Flag=0 Len=8 (4 bytes)
<47_54_36>
  # Transaction ID (8 bytes)
  1.33.830

# Header: Type=5A (Z) Flag=1 Len=126 (4 bytes)

```

The index redo is not extracted into the trail.

The commit redo is not extracted explicitly into the trail.

The total number of bytes required in the GoldenGate trail for the above change is 126.

The primary key value is not explicitly stored in the redo change vector; in this case it is included in the undo change vector.

Multi-Row Updates

The following statements were executed to prepare the STUDENT table for this example:

```
UPDATE student SET tuition_fee = 8000
WHERE student_key IN (1007,1008,1009);

COMMIT;
```

The purpose of executing these statements is to ensure that only three rows in the table have a TUITION_FEE value of 8000.

Consider the following statements:

```
UPDATE student SET tuition_fee = 7500
WHERE tuition_fee = 8000;

COMMIT;
```

The above statement sets the TUITION_FEE column to 7500 for the three rows where the TUITION_FEE is currently 8000. Note that the statement does not reference the rows by their primary key values.

The above statement generates the following entries in the redo log. The first redo record updates the first row to be modified into the STUDENT table:

```
REDO RECORD - Thread:1 RBA: 0x000047.00000006.0010 LEN: 0x0244 VLD: 0x05
SCN: 0x0000.0018e18e SUBSCN: 1 04/01/2013 05:14:37
(LWN RBA: 0x000047.00000006.0010 LEN: 0003 NST: 0001 SCN: 0x0000.0018e18d)

CHANGE #1 TYP:0 CLS:27 AFN:3 DBA:0x00c000d0 OBJ:4294967295 SCN:0x0000.0018e162
SEQ:1 OP:5.2 ENC:0 RBL:0
ktudh redo: slt: 0x001b sqn: 0x000004b4 flg: 0x0012 siz: 200 fbi: 0
           uba: 0x00c0029f.0119.1b pxid: 0x0000.000.00000000

CHANGE #2 TYP:0 CLS:28 AFN:3 DBA:0x00c0029f OBJ:4294967295 SCN:0x0000.0018e161
SEQ:2 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 200 spc: 3546 flg: 0x0012 seq: 0x0119 rec: 0x1b
           xid: 0x0006.01b.000004b4
ktubl redo: slt: 27 rci: 0 opc: 11.1 [objn: 76490 objd: 76495 tsn: 4]
Undo type: Regular undo Begin trans Last buffer split: No
Temp Object: No
Tablespace Undo: No
           0x00000000 prev ctl uba: 0x00c0029f.0119.19
prev ctl max cmt scn: 0x0000.0018d428 prev tx cmt scn: 0x0000.0018d43c
txn start scn: 0xffff.ffffffff logon user: 90 prev brb: 12583611 prev bcl: 0
BuExt idx: 0 flg2: 0
KDO undo record:
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0001.021.0000033e uba: 0x00c00157.01da.05
           flg: C--- lkc: 0 scn: 0x0000.0018cf25
KDO Op code: URP row dependencies Disabled
xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 6(0x6) flag: 0x2c lock: 0 ckix: 0
ncol: 8 nnew: 1 size: 0
col 7: [ 2] c2 51
```

```

CHANGE #3 TYP:2 CLS:1 AFN:4 DBA:0x01000436 OBJ:76495 SCN:0x0000.0018e0d8 SEQ:1
OP:11.5 ENC:0 RBL:0
KTB Redo
op: 0x11 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0006.01b.000004b4 uba: 0x00c0029f.0119.1b
Block cleanout record, scn: 0x0000.0018e18e ver: 0x01 opt: 0x02, entries follow...
itli: 1 flg: 2 scn: 0x0000.0018e0d8
KDO Op code: URP row dependencies Disabled
xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 6(0x6) flag: 0x2c lock: 2 ckix: 0
ncol: 8 nnew: 1 size: 0
col 7: [ 2] c2 4c

CHANGE #4 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:5.20 ENC:0
session number = 53
serial number = 61
transaction name =
version 186647296
audit sessionid 220222
Client Id =
login username = US03

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	68
1	5.2	Allocate transaction slot in undo header	60
2	5.1	Undo for update operation (URP)	248
3	11.5	Redo for update operation (URP)	140
4	5.20	Media recovery marker	64

In this example, no indexes are updated as the TUITION_FEE column is not included in any indexes.

The second redo record updates the second row to be modified into the STUDENT table:

```

REDO RECORD - Thread:1 RBA: 0x000047.00000007.0064 LEN: 0x0128 VLD: 0x01
SCN: 0x0000.0018e18e SUBSCN: 1 04/01/2013 05:14:37

CHANGE #1 TYP:0 CLS:28 AFN:3 DBA:0x00c0029f OBJ:4294967295 SCN:0x0000.0018e18e
SEQ:1 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 132 spc: 3344 flg: 0x0022 seq: 0x0119 rec: 0x1c
xid: 0x0006.01b.000004b4
ktubu redo: slt: 27 rci: 27 opc: 11.1 objn: 76490 objd: 76495 tsn: 4
Undo type: Regular undo Undo type: Last buffer split: No
Tablespace Undo: No
0x00000000
KDO undo record:
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c0029f.0119.1b
KDO Op code: URP row dependencies Disabled
xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 7(0x7) flag: 0x2c lock: 0 ckix: 0
ncol: 8 nnew: 1 size: 0
col 7: [ 2] c2 51

CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x01000436 OBJ:76495 SCN:0x0000.0018e18e SEQ:1
OP:11.5 ENC:0 RBL:0
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1

```



```

op: C uba: 0x00c0029f.0119.1c
KDO Op code: URP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 7(0x7) flag: 0x2c lock: 2 ckix: 0
ncol: 8 nnew: 1 size: 0
col 7: [ 2] c2 4c

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.1	Undo for update operation (URP)	180
2	11.5	Redo for update operation (URP)	92

The third redo record updates the third row to be modified into the STUDENT table:

```

REDO RECORD - Thread:1 RBA: 0x000047.00000007.018c LEN: 0x0128 VLD: 0x01
SCN: 0x0000.0018e18e SUBSCN: 1 04/01/2013 05:14:37
CHANGE #1 TYP:0 CLS:28 AFN:3 DBA:0x00c0029f OBJ:4294967295 SCN:0x0000.0018e18e
SEQ:2 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 132 spc: 3210 flg: 0x0022 seq: 0x0119 rec: 0x1d
  xid: 0x0006.01b.000004b4
ktubu redo: slt: 27 rci: 28 opc: 11.1 objn: 76490 objd: 76495 tsn: 4
Undo type: Regular undo Undo type: Last buffer split: No
Tablespace Undo: No
  0x00000000
KDO undo record:
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c0029f.0119.1c
KDO Op code: URP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 8(0x8) flag: 0x2c lock: 0 ckix: 0
ncol: 8 nnew: 1 size: 0
col 7: [ 2] c2 51
CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x01000436 OBJ:76495 SCN:0x0000.0018e18e SEQ:2
OP:11.5 ENC:0 RBL:0
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c0029f.0119.1d
KDO Op code: URP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 8(0x8) flag: 0x2c lock: 2 ckix: 0
ncol: 8 nnew: 1 size: 0
col 7: [ 2] c2 4c

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.1	Undo for update operation (URP)	180
2	11.5	Redo for update operation (URP)	92

The COMMIT statement generates the following redo:

```

REDO RECORD - Thread:1 RBA: 0x000047.00000008.00c4 LEN: 0x00a4 VLD: 0x01
SCN: 0x0000.0018e18f SUBSCN: 1 04/01/2013 05:14:37

```

```
CHANGE #1 TYP:0 CLS:27 AFN:3 DBA:0x00c000d0 OBJ:4294967295 SCN:0x0000.0018e18e
SEQ:1 OP:5.4 ENC:0 RBL:0
ktucm redo: slt: 0x001b sqn: 0x000004b4 srt: 0 sta: 9 flg: 0x2 ktucf redo: uba:
0x00c0029f.0119.1d ext: 2 spc: 3076 fbi: 0
CHANGE #2 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:24.4 ENC:0
```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.4	Release transaction slot in undo header	72
2	24.4	Media recovery marker	68

The redo log entries generate the following TrailAnalyzer output from the GoldenGate extract trail:

```
# Header: Type=47 (G) Flag=1 Len=139 (4 bytes)

# Row (131 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
# Row Header (47 bytes)
# Flags - 0x0F Update
# Timestamp 2013:04:01 05:14:37 000000 ms
# Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=24 (4 bytes)
<47_44>
# Row Data (24 bytes)
# Slot 0 (8 bytes)
00 00 00 04 31 30 30 37 # STUDENT_KEY = 1007
# Slot 7 (8 bytes)
00 00 00 04 37 35 30 30 # TUITION_FEE = 7500

# Header: Type=54 (T) Flag=0 Len=48 (4 bytes)
<47_54>
# Row Metadata (48 bytes)
# Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
<47_54_52>
# ROWID: 20 bytes
AAASrPAAEAAAAQ2AAG 1

# Header: Type=4C (L) Flag=0 Len=7 (4 bytes)
<47_54_4C>
# Commit SCN (7 bytes)
1630607

# Header: Type=36 (6) Flag=0 Len=9 (4 bytes)
<47_54_36>
# Transaction ID (9 bytes)
6.27.1204

# Header: Type=5A (Z) Flag=1 Len=139 (4 bytes)
# Header: Type=47 (G) Flag=1 Len=115 (4 bytes)

# Row (107 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
# Row Header (47 bytes)
# Flags - 0x0F Update
# Timestamp 2013:04:01 05:14:37 000000 ms
# Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=24 (4 bytes)
<47_44>
# Row Data (24 bytes)
```

```

# Slot 0 (8 bytes)
00 00 00 04 31 30 30 38          # STUDENT_KEY = 1008
# Slot 7 (8 bytes)
00 00 00 04 37 35 30 30          # TUITION_FEE = 7500

# Header: Type=54 (T) Flag=0 Len=24 (4 bytes)
<47_54>
# Row Metadata (24 bytes)
# Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
<47_54_52>
# ROWID: 20 bytes
AAASrPAAEAAAAQ2AAH 1

# Header: Type=5A (Z) Flag=1 Len=115 (4 bytes)
# Header: Type=47 (G) Flag=1 Len=115 (4 bytes)

# Row (107 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
# Row Header (47 bytes)
# Flags - 0x0F Update
# Timestamp 2013:04:01 05:14:37 000000 ms
# Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=24 (4 bytes)
<47_44>
# Row Data (24 bytes)
# Slot 0 (8 bytes)
00 00 00 04 31 30 30 39          # STUDENT_KEY = 1009
# Slot 7 (8 bytes)
00 00 00 04 37 35 30 30          # TUITION_FEE = 7500

# Header: Type=54 (T) Flag=0 Len=24 (4 bytes)
<47_54>
# Row Metadata (24 bytes)
# Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
<47_54_52>
# ROWID: 20 bytes
AAASrPAAEAAAAQ2AAI 1

# Header: Type=5A (Z) Flag=1 Len=115 (4 bytes)

```

The index redo is not extracted into the trail.

The commit redo is not extracted explicitly into the trail.

The total number of bytes required in the GoldenGate trail for the above change is 369.

Note that the primary keys are included in the trail. These are required to identify the rows in the target database. Although the ROWIDs are included in the trail, they cannot be used to locate the row in the target database as the ROWIDs can differ between the source and target databases.

The primary key values are not explicitly stored in the redo change vectors; they are included in the supplemental data for the undo change vector.

Multi-Row Deletes

The following statements were executed to prepare the STUDENT table for this example:

```

UPDATE student SET tuition_fee = 8000
WHERE student_key IN (1007,1008,1009);

COMMIT;

```

The purpose of executing these statements is to ensure that only three rows in the table have a TUITION_FEE value of 8000.

Consider the following statements:

```
DELETE FROM student WHERE tuition_fee = 8000;

COMMIT;
```

The above statement deletes all rows where the TUITION_FEE is currently 8000. Note that the statement does not reference the rows by their primary key values.

The above statement generates the following entries in the redo log. The first redo record deletes the first row from the STUDENT table:

```
REDO RECORD - Thread:1 RBA: 0x000049.00000007.0010 LEN: 0x0268 VLD: 0x05
SCN: 0x0000.0018ffde SUBSCN: 1 04/01/2013 09:49:28
(LWN RBA: 0x000049.00000007.0010 LEN: 0006 NST: 0001 SCN: 0x0000.0018ffdd)

CHANGE #1 TYP:0 CLS:21 AFN:3 DBA:0x00c000a0 OBJ:4294967295 SCN:0x0000.0018ff5a
SEQ:1 OP:5.2 ENC:0 RBL:0
ktudh redo: slt: 0x0017 sqn: 0x000003f8 flg: 0x0012 siz: 256 fbi: 0
          uba: 0x00c000a4.015f.15      pxid: 0x0000.000.00000000

CHANGE #2 TYP:0 CLS:22 AFN:3 DBA:0x00c000a4 OBJ:4294967295 SCN:0x0000.0018ff59
SEQ:2 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 256 spc: 3450 flg: 0x0012 seq: 0x015f rec: 0x15
          xid: 0x0003.017.000003f8
ktubl redo: slt: 23 rci: 0 opc: 11.1 [objn: 76490 objd: 76495 tsn: 4]
Undo type: Regular undo          Begin trans      Last buffer split: No
Temp Object: No
Tablespace Undo: No
          0x00000000 prev ctl uba: 0x00c000a4.015f.13
prev ctl max cmt scn: 0x0000.0018f39c prev tx cmt scn: 0x0000.0018f3ae
txn start scn: 0xffff.ffffffff logon user: 90 prev brb: 12584363 prev bcl: 0
BuExt idx: 0 flg2: 0
KDO undo record:
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0009.01a.0000040e uba: 0x00c00127.01bb.04
          flg: C--- lkc: 0      scn: 0x0000.0018f237
KDO Op code: IRP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 3(0x3) size/delt: 47
fb: --H-FL-- lb: 0x0 cc: 8
null: -----
col 0: [ 3] c2 0b 08
col 1: [ 8] 56 69 63 74 6f 72 69 61
col 2: [ 5] 45 76 61 6e 73
col 3: [ 1] 46
col 4: [ 6] 4f 78 66 6f 72 64
col 5: [ 8] 54 68 65 6f 6c 6f 67 79
col 6: [ 3] c2 15 0e
col 7: [ 2] c2 51

CHANGE #3 TYP:2 CLS:1 AFN:4 DBA:0x01000436 OBJ:76495 SCN:0x0000.0018ff5c SEQ:1
OP:11.3 ENC:0 RBL:0
KTB Redo
op: 0x11 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0003.017.000003f8 uba: 0x00c000a4.015f.15
Block cleanout record, scn: 0x0000.0018ffde ver: 0x01 opt: 0x02, entries follow...
  itli: 1 flg: 2 scn: 0x0000.0018ff5c
  itli: 2 flg: 2 scn: 0x0000.0018f237
```

```

KDO Op code: DRP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 3(0x3)

CHANGE #4 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:5.20 ENC:0
session number = 42
serial number = 1179
transaction name =
version 186647296
audit sessionid 220277
Client Id =
login username = US03

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	68
1	5.2	Allocate transaction slot in undo header	60
2	5.1	Undo for delete operation (IRP)	300
3	11.3	Redo for delete operation (DRP)	124
4	5.20	Media recovery marker	64

The second redo record deletes the index entry for the first row from the STUDENT_PK index:

```

REDO RECORD - Thread:1 RBA: 0x000049.00000008.00d0 LEN: 0x00ec VLD: 0x01
SCN: 0x0000.0018ffde SUBSCN: 1 04/01/2013 09:49:28

CHANGE #1 TYP:0 CLS:22 AFN:3 DBA:0x00c000a4 OBJ:4294967295 SCN:0x0000.0018ffde
SEQ:1 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 100 spc: 3192 flg: 0x0022 seq: 0x015f rec: 0x16
          xid: 0x0003.017.000003f8
ktubu redo: slt: 23 rci: 21 opc: 10.22 objn: 76491 objd: 76494 tsn: 4
Undo type: Regular undo          Undo type: Last buffer split: No
Tablespace Undo: No
          0x00000000
index undo for leaf key operations
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0006.013.000004b6 uba: 0x00c002d7.0119.1f
          flg: C--- lkc: 0          scn: 0x0000.0018ff5c
Dump kdilk : itl=2, kdxlklflg=0x1 sdc=0 indexid=0x100043a block=0x0100043b
(kdxlre): restore leaf row (clear leaf delete flags)
key : (4): 03 c2 0b 08
keydata/bitmap: (6): 01 00 04 36 00 03

CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x0100043b OBJ:76494 SCN:0x0000.0018ffde SEQ:1
OP:10.4 ENC:0 RBL:0
index redo (kdxlde): delete leaf row
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0003.017.000003f8          uba: 0x00c000a4.015f.16
REDO: SINGLE / -- / --
itl: 2, sno: 5, row size 14

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.1	Undo for delete index row	148
2	10.4	Redo for delete index row	64

The third redo record deletes the index entry for the first row from the STUDENT_UK index:

```

REDO RECORD - Thread:1 RBA: 0x000049.00000009.0014 LEN: 0x00f8 VLD: 0x01
SCN: 0x0000.0018ffde SUBSCN: 1 04/01/2013 09:49:28

CHANGE #1 TYP:0 CLS:22 AFN:3 DBA:0x00c000a4 OBJ:4294967295 SCN:0x0000.0018ffde
SEQ:2 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 112 spc: 3090 flg: 0x0022 seq: 0x015f rec: 0x17
            xid: 0x0003.017.000003f8
ktubu redo: slt: 23 rci: 22 opc: 10.22 objn: 76492 objd: 76493 tsn: 4
Undo type: Regular undo          Undo type: Last buffer split: No
Tablespace Undo: No
            0x00000000
index undo for leaf key operations
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0006.013.000004b6 uba: 0x00c002d7.0119.20
            flg: C-- lkc: 0 scn: 0x0000.0018ff5c
Dump kdilk : itl=2, kdxlkflg=0x1 sdc=0 indexid=0x1000342 block=0x01000343
(kdxmlre): restore leaf row (clear leaf delete flags)
key : (15): 08 56 69 63 74 6f 72 69 61 05 45 76 61 6e 73
keydata/bitmap: (6): 01 00 04 36 00 03

CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x01000343 OBJ:76493 SCN:0x0000.0018ffde SEQ:1
OP:10.4 ENC:0 RBL:0
index redo (kdxlde): delete leaf row
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0003.017.000003f8 uba: 0x00c000a4.015f.17
REDO: SINGLE / -- / --
itl: 2, sno: 9, row size 25

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.1	Undo for delete index row	160
2	10.4	Redo for delete index row	64

The fourth redo record deletes the second row from the STUDENT table:

```

REDO RECORD - Thread:1 RBA: 0x000049.00000009.010c LEN: 0x0140 VLD: 0x01
SCN: 0x0000.0018ffde SUBSCN: 1 04/01/2013 09:49:28

CHANGE #1 TYP:0 CLS:22 AFN:3 DBA:0x00c000a4 OBJ:4294967295 SCN:0x0000.0018ffde
SEQ:3 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 184 spc: 2976 flg: 0x0022 seq: 0x015f rec: 0x18
            xid: 0x0003.017.000003f8
ktubu redo: slt: 23 rci: 23 opc: 11.1 objn: 76490 objd: 76495 tsn: 4
Undo type: Regular undo          Undo type: Last buffer split: No
Tablespace Undo: No
            0x00000000
KDO undo record:
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c000a4.015f.15
KDO Op code: IRP row dependencies Disabled
xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 11(0xb) size/delt: 44
fb: --H-FL-- lb: 0x0 cc: 8
null: -----

```

```

col 0: [ 3] c2 0b 09
col 1: [ 4] 4b 61 74 79
col 2: [ 6] 50 69 65 72 63 65
col 3: [ 1] 46
col 4: [ 6] 4f 78 66 6f 72 64
col 5: [ 8] 54 68 65 6f 6c 6f 67 79
col 6: [ 3] c2 15 0e
col 7: [ 2] c2 51

CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x01000436 OBJ:76495 SCN:0x0000.0018ffde SEQ:1
OP:11.3 ENC:0 RBL:0
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c000a4.015f.18
KDO Op code: DRP row dependencies Disabled
xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 11(0xb)

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.1	Undo for delete operation (IRP)	228
2	11.3	Redo for delete operation (DRP)	68

The fifth redo record deletes the index entry for the second row from the STUDENT_PK index:

```

REDO RECORD - Thread:1 RBA: 0x000049.0000000a.005c LEN: 0x00d4 VLD: 0x01
SCN: 0x0000.0018ffde SUBSCN: 1 04/01/2013 09:49:28

CHANGE #1 TYP:0 CLS:22 AFN:3 DBA:0x00c000a4 OBJ:4294967295 SCN:0x0000.0018ffde
SEQ:4 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 84 spc: 2790 flg: 0x0022 seq: 0x015f rec: 0x19
xid: 0x0003.017.000003f8
ktubu redo: slt: 23 rci: 24 opc: 10.22 objn: 76491 objd: 76494 tsn: 4
Undo type: Regular undo Undo type: Last buffer split: No
Tablespace Undo: No
0x00000000
index undo for leaf key operations
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c000a4.015f.16
Dump kdilk : itl=2, kdxlklfg=0x1 sdc=0 indexid=0x100043a block=0x0100043b
(kdxlre): restore leaf row (clear leaf delete flags)
key : (4): 03 c2 0b 09
keydata/bitmap: (6): 01 00 04 36 00 0b

CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x0100043b OBJ:76494 SCN:0x0000.0018ffde SEQ:2
OP:10.4 ENC:0 RBL:0
index redo (kdxlde): delete leaf row
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c000a4.015f.19
REDO: SINGLE / -- / --
itl: 2, sno: 6, row size 14

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24

1	5.1	Undo for delete index row	132
2	10.4	Redo for delete index row	64

The sixth redo record deletes the index entry for the second row from the STUDENT_UK index:

```

REDO RECORD - Thread:1 RBA: 0x000049.0000000a.0130 LEN: 0x00dc VLD: 0x01
SCN: 0x0000.0018ffde SUBSCN: 1 04/01/2013 09:49:28

CHANGE #1 TYP:0 CLS:22 AFN:3 DBA:0x00c000a4 OBJ:4294967295 SCN:0x0000.0018ffde
SEQ:5 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 92 spc: 2704 flg: 0x0022 seq: 0x015f rec: 0x1a
          xid: 0x0003.017.000003f8
ktubu redo: slt: 23 rci: 25 opc: 10.22 objn: 76492 objd: 76493 tsn: 4
Undo type: Regular undo          Undo type: Last buffer split: No
Tablespace Undo: No
          0x00000000
index undo for leaf key operations
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c000a4.015f.17
Dump kdilk : itl=2, kdxlkflg=0x1 sdc=0 indexid=0x1000342 block=0x01000343
(kdxlre): restore leaf row (clear leaf delete flags)
key :(12): 04 4b 61 74 79 06 50 69 65 72 63 65
keydata/bitmap: (6): 01 00 04 36 00 0b

CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x01000343 OBJ:76493 SCN:0x0000.0018ffde SEQ:2
OP:10.4 ENC:0 RBL:0
index redo (kdxlde): delete leaf row
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c000a4.015f.1a
REDO: SINGLE / -- / --
itl: 2, sno: 1, row size 22

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.1	Undo for delete index row	140
2	10.4	Redo for delete index row	64

The seventh redo record deletes the third row from the STUDENT table:

```

REDO RECORD - Thread:1 RBA: 0x000049.0000000b.001c LEN: 0x0150 VLD: 0x01
SCN: 0x0000.0018ffde SUBSCN: 1 04/01/2013 09:49:28

CHANGE #1 TYP:0 CLS:22 AFN:3 DBA:0x00c000a4 OBJ:4294967295 SCN:0x0000.0018ffde
SEQ:6 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 200 spc: 2610 flg: 0x0022 seq: 0x015f rec: 0x1b
          xid: 0x0003.017.000003f8
ktubu redo: slt: 23 rci: 26 opc: 11.1 objn: 76490 objd: 76495 tsn: 4
Undo type: Regular undo          Undo type: Last buffer split: No
Tablespace Undo: No
          0x00000000
KDO undo record:
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c000a4.015f.18
KDO Op code: IRP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858

```



```

tabn: 0 slot: 12(0xc) size/delt: 54
fb: --H-FL-- lb: 0x0 cc: 8
null: -----
col 0: [ 3] c2 0b 0a
col 1: [ 5] 53 68 61 6e 65
col 2: [ 6] 54 68 6f 6d 61 73
col 3: [ 1] 4d
col 4: [10] 4d 61 6e 63 68 65 73 74 65 72
col 5: [13] 4d 65 64 69 61 20 53 74 75 64 69 65 73
col 6: [ 3] c2 15 0e
col 7: [ 2] c2 51

CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x01000436 OBJ:76495 SCN:0x0000.0018ffde SEQ:2
OP:11.3 ENC:0 RBL:0
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c000a4.015f.1b
KDO Op code: DRP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 2 ispac: 0 maxfr: 4858
tabn: 0 slot: 12(0xc)

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.1	Undo for delete operation (IRP)	244
2	11.3	Redo for delete operation (DRP)	68

The eighth redo record deletes the index entry for the third row from the STUDENT_PK index:

```

REDO RECORD - Thread:1 RBA: 0x000049.0000000b.016c LEN: 0x00d4 VLD: 0x01
SCN: 0x0000.0018ffde SUBSCN: 1 04/01/2013 09:49:28

CHANGE #1 TYP:0 CLS:22 AFN:3 DBA:0x00c000a4 OBJ:4294967295 SCN:0x0000.0018ffde
SEQ:7 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 84 spc: 2408 flg: 0x0022 seq: 0x015f rec: 0x1c
  xid: 0x0003.017.000003f8
ktubu redo: slt: 23 rci: 27 opc: 10.22 objn: 76491 objd: 76494 tsn: 4
Undo type: Regular undo Undo type: Last buffer split: No
Tablespace Undo: No
  0x00000000
index undo for leaf key operations
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c000a4.015f.19
Dump kdilk : itl=2, kdxlkflg=0x1 sdc=0 indexid=0x100043a block=0x0100043b
(kdxlre): restore leaf row (clear leaf delete flags)
key : (4): 03 c2 0b 0a
keydata/bitmap: (6): 01 00 04 36 00 0c

CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x0100043b OBJ:76494 SCN:0x0000.0018ffde SEQ:3
OP:10.4 ENC:0 RBL:0
index redo (kdxlde): delete leaf row
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c000a4.015f.1c
REDO: SINGLE / -- / --
itl: 2, sno: 7, row size 14

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
2	5.1	Undo for delete index row	132
3	10.4	Redo for delete index row	64

The ninth redo record deletes the index entry for the third row from the STUDENT_UK index:

```

REDO RECORD - Thread:1 RBA: 0x000049.0000000c.0050 LEN: 0x00e0 VLD: 0x01
SCN: 0x0000.0018ffde SUBSCN: 1 04/01/2013 09:49:28

CHANGE #1 TYP:0 CLS:22 AFN:3 DBA:0x00c000a4 OBJ:4294967295 SCN:0x0000.0018ffde
SEQ:8 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 96 spc: 2322 flg: 0x0022 seq: 0x015f rec: 0x1d
          xid: 0x0003.017.000003f8
ktubu redo: slt: 23 rci: 28 opc: 10.22 objn: 76492 objd: 76493 tsn: 4
Undo type: Regular undo          Undo type: Last buffer split: No
Tablespace Undo: No
          0x00000000
index undo for leaf key operations
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c000a4.015f.1a
Dump kdilk : itl=2, kdxlkflg=0x1 sdc=0 indexid=0x1000342 block=0x01000343
(kdxlre): restore leaf row (clear leaf delete flags)
key : (13): 05 53 68 61 6e 65 06 54 68 6f 6d 61 73
keydata/bitmap: (6): 01 00 04 36 00 0c

CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x01000343 OBJ:76493 SCN:0x0000.0018ffde SEQ:3
OP:10.4 ENC:0 RBL:0
index redo (kdxlde): delete leaf row
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x00c000a4.015f.1d
REDO: SINGLE / -- / --
itl: 2, sno: 5, row size 23

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
2	5.1	Undo for delete index row	144
3	10.4	Redo for delete index row	64

The COMMIT statement generates the following redo:

```

REDO RECORD - Thread:1 RBA: 0x000049.0000000c.0130 LEN: 0x00a4 VLD: 0x01
SCN: 0x0000.0018ffdf SUBSCN: 1 04/01/2013 09:49:28
CHANGE #1 TYP:0 CLS:21 AFN:3 DBA:0x00c000a0 OBJ:4294967295 SCN:0x0000.0018ffde
SEQ:1 OP:5.4 ENC:0 RBL:0
ktucm redo: slt: 0x0017 sqn: 0x000003f8 srt: 0 sta: 9 flg: 0x2 ktucf redo: uba:
0x00c000a4.015f.1d ext: 0 spc: 2224 fbi: 0
CHANGE #2 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:24.4 ENC:0

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.4	Release transaction slot in undo header	72
2	24.4	Media recovery marker	68

The redo log entries generate the following TrailAnalyzer output from the GoldenGate extract trail:

The following is added to the GoldenGate trail for the first row updated.

```
# Header: Type=47 (G) Flag=1 Len=127 (4 bytes)

# Row (119 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
  # Row Header (47 bytes)
  # Flags - 0x03 Delete
  # Timestamp 2013:04:01 09:49:28 000000 ms
  # Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=12 (4 bytes)
<47_44>
  # Row Data (12 bytes)
  # Slot 0 (8 bytes)
  00 00 00 04 31 30 30 37

# Header: Type=54 (T) Flag=0 Len=48 (4 bytes)
<47_54>
  # Row Metadata (48 bytes)
  # Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
  <47_54_52>
  # ROWID: 20 bytes
  AAASrPAAEAAAAQ2AAD 1

  # Header: Type=4C (L) Flag=0 Len=7 (4 bytes)
  <47_54_4C>
  # Commit SCN (7 bytes)
  1638367

  # Header: Type=36 (6) Flag=0 Len=9 (4 bytes)
  <47_54_36>
  # Transaction ID (9 bytes)
  3.23.1016

# Header: Type=5A (Z) Flag=1 Len=127 (4 bytes)
```

The following is added to the GoldenGate trail for the second row updated.

```
# Header: Type=47 (G) Flag=1 Len=103 (4 bytes)

# Row (95 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
  # Row Header (47 bytes)
  # Flags - 0x03 Delete
  # Timestamp 2013:04:01 09:49:28 000000 ms
  # Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=12 (4 bytes)
<47_44>
  # Row Data (12 bytes)
  # Slot 0 (8 bytes)
  00 00 00 04 31 30 30 38

# Header: Type=54 (T) Flag=0 Len=24 (4 bytes)
<47_54>
  # Row Metadata (24 bytes)
  # Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
  <47_54_52>
  # ROWID: 20 bytes
  AAASrPAAEAAAAQ2AAL 1
```

```
# Header: Type=5A (Z) Flag=1 Len=103 (4 bytes)
```

The following is added to the GoldenGate trail for the thirdcd row updated.

```
# Header: Type=47 (G) Flag=1 Len=103 (4 bytes)

# Row (95 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
  # Row Header (47 bytes)
  # Flags - 0x03 Delete
  # Timestamp 2013:04:01 09:49:28 000000 ms
  # Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=12 (4 bytes)
<47_44>
  # Row Data (12 bytes)
  # Slot 0 (8 bytes)
  00 00 00 04 31 30 30 39

# Header: Type=54 (T) Flag=0 Len=24 (4 bytes)
<47_54>
  # Row Metadata (24 bytes)
  # Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
  <47_54_52>
  # ROWID: 20 bytes
  AAASrPAAEAAAAQ2AAM 1

# Header: Type=5A (Z) Flag=1 Len=103 (4 bytes)
```

The index redo is not extracted into the trail.

The commit redo is not extracted explicitly into the trail.

The total number of bytes required in the GoldenGate trail for the above change is 333.

Note that the primary keys are included in the trail. These are required to identify the rows in the target database. Although the ROWIDs are included in the trail, they cannot be used to locate the row in the target database as the ROWIDs can differ between the source and target databases.

The primary key values are not explicitly stored in the redo change vectors; they are included in the supplemental data for the undo change vector.

The primary key values are not explicitly stored in the redo change vector; however they are included in the undo change vector.

Array Inserts

The following statements were executed to prepare the STUDENT table for this example:

```
CREATE TABLE student_temp AS SELECT * FROM student WHERE ROWNUM < 1;

INSERT INTO student_temp VALUES (1007,'Victoria','Evans','F',
'OXford','Theology',2013,9000);

INSERT INTO student_temp VALUES (1008,'Katy','Pierce','F',
'OXford','Theology',2013,9000);

INSERT INTO student_temp VALUES (1009,'Shane','Thomas','M',
'MANCHESTER','Media Studies',2013,9000);

DELETE FROM student WHERE student_key IN (1007,1008,1009);

COMMIT;
```

Consider the following statements:

```
INSERT INTO student SELECT * FROM student_temp;

COMMIT;
```

The above statement generates the following entries in the redo log. The first redo record inserts an array of rows into the STUDENT table:

```
REDO RECORD - Thread:1 RBA: 0x00004b.00000005.0010 LEN: 0x0288 VLD: 0x05
SCN: 0x0000.00190cd2 SUBSCN: 1 04/01/2013 11:38:17
(LWN RBA: 0x00004b.00000005.0010 LEN: 0004 NST: 0001 SCN: 0x0000.00190cd2)

CHANGE #1 TYP:0 CLS:29 AFN:3 DBA:0x00c000e0 OBJ:4294967295 SCN:0x0000.00190cac
SEQ:1 OP:5.2 ENC:0 RBL:0
ktudh redo: slt: 0x000d sqn: 0x0000034e flg: 0x0012 siz: 168 fbi: 0
           uba: 0x00c00767.016b.14      pxid: 0x0000.000.00000000

CHANGE #2 TYP:0 CLS:30 AFN:3 DBA:0x00c00767 OBJ:4294967295 SCN:0x0000.00190cab
SEQ:2 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 168 spc: 4876 flg: 0x0012 seq: 0x016b rec: 0x14
           xid: 0x0007.00d.0000034e
ktubl redo: slt: 13 rci: 0 opc: 11.1 [objn: 76490 objd: 76495 tsn: 4]
Undo type: Regular undo          Begin trans      Last buffer split: No
Temp Object: No
Tablespace Undo: No
           0x00000000 prev ctl uba: 0x00c00767.016b.12
prev ctl max cmt scn: 0x0000.00190124 prev tx cmt scn: 0x0000.00190167
txn start scn: 0xffff.ffffffff logon user: 90 prev brb: 12584806 prev bcl: 0
BuExt idx: 0 flg2: 0
KDO undo record:
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0006.013.000004b6 uba: 0x00c002d7.0119.1e
           flg: C--- lkc: 0      scn: 0x0000.0018ff5c
KDO Op code: QMD row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 1 ispac: 0 maxfr: 4858
tabn: 0 lock: 0 nrow: 3
slot[0]: 6
slot[1]: 7
slot[2]: 8

CHANGE #3 TYP:2 CLS:1 AFN:4 DBA:0x01000436 OBJ:76495 SCN:0x0000.0018ffdf SEQ:1
OP:11.11 ENC:0 RBL:0
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0007.00d.0000034e uba: 0x00c00767.016b.14
KDO Op code: QMI row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x01000436 hdba: 0x01000432
itli: 1 ispac: 0 maxfr: 4858
tabn: 0 lock: 1 nrow: 3
slot[0]: 6
tl: 47 fb: --H-FL-- lb: 0x0 cc: 8
col 0: [ 3] c2 0b 08
col 1: [ 8] 56 69 63 74 6f 72 69 61
col 2: [ 5] 45 76 61 6e 73
col 3: [ 1] 46
col 4: [ 6] 4f 78 66 6f 72 64
col 5: [ 8] 54 68 65 6f 6c 6f 67 79
col 6: [ 3] c2 15 0e
col 7: [ 2] c2 5b
slot[1]: 7
tl: 44 fb: --H-FL-- lb: 0x0 cc: 8
col 0: [ 3] c2 0b 09
```

```

col 1: [ 4] 4b 61 74 79
col 2: [ 6] 50 69 65 72 63 65
col 3: [ 1] 46
col 4: [ 6] 4f 78 66 6f 72 64
col 5: [ 8] 54 68 65 6f 6c 6f 67 79
col 6: [ 3] c2 15 0e
col 7: [ 2] c2 5b
slot[2]: 8
tl: 54 fb: --H-FL-- lb: 0x0 cc: 8
col 0: [ 3] c2 0b 0a
col 1: [ 5] 53 68 61 6e 65
col 2: [ 6] 54 68 6f 6d 61 73
col 3: [ 1] 4d
col 4: [10] 4d 61 6e 63 68 65 73 74 65 72
col 5: [13] 4d 65 64 69 61 20 53 74 75 64 69 65 73
col 6: [ 3] c2 15 0e
col 7: [ 2] c2 5b

CHANGE #4 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:5.20 ENC:0
session number = 44
serial number = 845
transaction name =
version 186647296
audit sessionid 220306
Client Id =
login username = US03

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	68
1	5.2	Allocate transaction slot in undo header	60
2	5.1	Undo for array insert operation (QMD)	212
3	11.3	Redo for array insert operation (QMI)	244
4	5.20	Media recovery marker	64

Note that the above redo inserts three rows in a single redo record. Array inserts are much more efficient in terms of redo than single row inserts.

The second redo record inserts an array of index entries into the STUDENT_PK index:

```

REDO RECORD - Thread:1 RBA: 0x00004b.00000006.00f0 LEN: 0x0134 VLD: 0x01
SCN: 0x0000.00190cd2 SUBSCN: 1 04/01/2013 11:38:17
CHANGE #1 TYP:0 CLS:30 AFN:3 DBA:0x00c00767 OBJ:4294967295 SCN:0x0000.00190cd2
SEQ:1 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 116 spc: 4706 flg: 0x0022 seq: 0x016b rec: 0x15
          xid: 0x0007.00d.0000034e
ktubu redo: slt: 13 rci: 20 opc: 10.22 objn: 76491 objd: 76494 tsn: 4
Undo type: Regular undo          Undo type: Last buffer split: No
Tablespace Undo: No
          0x00000000
index undo for leaf key operations
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0003.017.000003f8 uba: 0x00c000a4.015f.1c
          flg: C--- lkc: 0 scn: 0x0000.0018ffdf
Dump kdilk : itl=2, kdxlklflg=0x21 sdc=0 indexid=0x100043a block=0x0100043b
(kdxlpu): purge leaf row
number of keys: 3
key sizes:
 4 4 4
key : (12): 03 c2 0b 08 03 c2 0b 09 03 c2 0b 0a
keydata/bitmap: (1): ff

```

```

CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x0100043b OBJ:76494 SCN:0x0000.00190cd2 SEQ:1
OP:10.2 ENC:0 RBL:0
index redo (kdxlin): insert leaf row
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0007.00d.0000034e uba: 0x00c00767.016b.15
REDO: ARRAY / -- / --
itl: 2, sno: 5, row size 42
number of keys: 3
slots:
 5 6 7
insert key: (12): 03 c2 0b 08 03 c2 0b 09 03 c2 0b 0a
each key size is:
 4 4 4
keydata: (18): 01 00 04 36 00 06 01 00 04 36 00 07 01 00 04 36 00 08

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.1	Undo for insert index rows	164
2	10.2	Redo for insert index rows	120

In this example, all the index rows can be stored on the index root block. Therefore only one block is affected by the change. If more than one block was affected by the above change, then separate redo records would be required for each block modified.

The third redo record inserts an array of index entries into the STUDENT_UK index:

```

REDO RECORD - Thread:1 RBA: 0x00004b.00000007.007c LEN: 0x016c VLD: 0x01
SCN: 0x0000.00190cd2 SUBSCN: 1 04/01/2013 11:38:17
CHANGE #1 TYP:0 CLS:30 AFN:3 DBA:0x00c00767 OBJ:4294967295 SCN:0x0000.00190cd2
SEQ:2 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 144 spc: 4588 flg: 0x0022 seq: 0x016b rec: 0x16
          xid: 0x0007.00d.0000034e
ktubu redo: slt: 13 rci: 21 opc: 10.22 objn: 76492 objd: 76493 tsn: 4
Undo type: Regular undo          Undo type: Last buffer split: No
Tablespace Undo: No
          0x00000000
index undo for leaf key operations
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0003.017.000003f8 uba: 0x00c000a4.015f.1d
          flg: C--- lkc: 0          scn: 0x0000.0018ffdf
Dump kdilk : itl=2, kdxlklflg=0x21 sdc=0 indexid=0x1000342 block=0x01000343
(kdxlpu): purge leaf row
number of keys: 3
key sizes:
 12 13 15
key : (40):
 04 4b 61 74 79 06 50 69 65 72 63 65 05 53 68 61 6e 65 06 54 68 6f 6d 61 73
 08 56 69 63 74 6f 72 69 61 05 45 76 61 6e 73
keydata/bitmap: (1): ff
CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x01000343 OBJ:76493 SCN:0x0000.00190cd2 SEQ:1
OP:10.2 ENC:0 RBL:0
index redo (kdxlin): insert leaf row
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0007.00d.0000034e uba: 0x00c00767.016b.16
REDO: ARRAY / -- / --
itl: 2, sno: 7, row size 70
number of keys: 3

```

```

slots:
 1 5 9
insert key: (40):
04 4b 61 74 79 06 50 69 65 72 63 65 05 53 68 61 6e 65 06 54 68 6f 6d 61 73
 08 56 69 63 74 6f 72 69 61 05 45 76 61 6e 73
each key size is:
 12 13 15
keydata: (18): 01 00 04 36 00 07 01 00 04 36 00 08 01 00 04 36 00 06

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.1	Undo for insert index rows	192
2	10.2	Redo for insert index rows	148

Again all the index rows can be stored on the index root block. Therefore only one block is affected by the change.

The COMMIT statement generates the following redo:

```

REDO RECORD - Thread:1 RBA: 0x00004b.00000007.01e8 LEN: 0x00a4 VLD: 0x01
SCN: 0x0000.00190cd3 SUBSCN: 1 04/01/2013 11:38:17
CHANGE #1 TYP:0 CLS:29 AFN:3 DBA:0x00c000e0 OBJ:4294967295 SCN:0x0000.00190cd2
SEQ:1 OP:5.4 ENC:0 RBL:0
ktucm redo: slt: 0x000d sqn: 0x0000034e srt: 0 sta: 9 flg: 0x2 ktucf redo: uba:
0x00c00767.016b.16 ext: 2 spc: 4442 fbi: 0
CHANGE #2 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:24.4 ENC:0

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.4	Release transaction slot in undo header	72
2	24.4	Media recovery marker	68

The redo log entries generate the following TrailAnalyzer output from the GoldenGate extract trail:

The following is added to the GoldenGate trail for the first row inserted.

```

# Header: Type=47 (G) Flag=1 Len=218 (4 bytes)

# Row (210 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
  # Row Header (47 bytes)
  # Flags - 0x05 Insert
  # Timestamp 2013:04:01 11:38:17 000000 ms
  # Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=104 (4 bytes)
<47_44>
  # Row Data (104 bytes)
  # Slot 0 (8 bytes)
  00 00 00 04 31 30 30 37
  # Slot 1 (12 bytes)
  00 00 00 08 56 69 63 74 6F 72 69 61
  # Slot 2 (9 bytes)
  00 00 00 05 45 76 61 6E 73
  # Slot 3 (5 bytes)
  00 00 00 01 46
  # Slot 4 (10 bytes)

```



```

00 00 00 06 4F 78 66 6F 72 64
# Slot 5 (12 bytes)
00 00 00 08 54 68 65 6F 6C 6F 67 79
# Slot 6 (8 bytes)
00 00 00 04 32 30 31 33
# Slot 7 (8 bytes)
00 00 00 04 39 30 30 30

# Header: Type=54 (T) Flag=0 Len=47 (4 bytes)
<47_54>
# Row Metadata (47 bytes)
# Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
<47_54_52>
# ROWID: 20 bytes
AAASrPAAEAAAAQ2AAG 1

# Header: Type=4C (L) Flag=0 Len=7 (4 bytes)
<47_54_4C>
# Commit SCN (7 bytes)
1641683

# Header: Type=36 (6) Flag=0 Len=8 (4 bytes)
<47_54_36>
# Transaction ID (8 bytes)
7.13.846

# Header: Type=5A (Z) Flag=1 Len=218 (4 bytes)

```

The following is added to the GoldenGate trail for the second row inserted.

```

# Header: Type=47 (G) Flag=1 Len=192 (4 bytes)

# Row (184 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
# Row Header (47 bytes)
# Flags - 0x05 Insert
# Timestamp 2013:04:01 11:38:17 000000 ms
# Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=101 (4 bytes)
<47_44>
# Row Data (101 bytes)
# Slot 0 (8 bytes)
00 00 00 04 31 30 30 38
# Slot 1 (8 bytes)
00 00 00 04 4B 61 74 79
# Slot 2 (10 bytes)
00 00 00 06 50 69 65 72 63 65
# Slot 3 (5 bytes)
00 00 00 01 46
# Slot 4 (10 bytes)
00 00 00 06 4F 78 66 6F 72 64
# Slot 5 (12 bytes)
00 00 00 08 54 68 65 6F 6C 6F 67 79
# Slot 6 (8 bytes)
00 00 00 04 32 30 31 33
# Slot 7 (8 bytes)
00 00 00 04 39 30 30 30

# Header: Type=54 (T) Flag=0 Len=24 (4 bytes)
<47_54>
# Row Metadata (24 bytes)
# Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
<47_54_52>
# ROWID: 20 bytes
AAASrPAAEAAAAQ2AAH 1

```

```
# Header: Type=5A (Z) Flag=1 Len=192 (4 bytes)
```

The following is added to the GoldenGate trail for the third row inserted.

```
# Header: Type=47 (G) Flag=1 Len=202 (4 bytes)

# Row (194 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
# Row Header (47 bytes)
# Flags - 0x05 Insert
# Timestamp 2013:04:01 11:38:17 000000 ms
# Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=111 (4 bytes)
<47_44>
# Row Data (111 bytes)
# Slot 0 (8 bytes)
00 00 00 04 31 30 30 39
# Slot 1 (9 bytes)
00 00 00 05 53 68 61 6E 65
# Slot 2 (10 bytes)
00 00 00 06 54 68 6F 6D 61 73
# Slot 3 (5 bytes)
00 00 00 01 4D
# Slot 4 (14 bytes)
00 00 00 0A 4D 61 6E 63 68 65 73 74 65 72
# Slot 5 (17 bytes)
00 00 00 0D 4D 65 64 69 61 20 53 74 75 64 69 65
73
# Slot 6 (8 bytes)
00 00 00 04 32 30 31 33
# Slot 7 (8 bytes)
00 00 00 04 39 30 30 30

# Header: Type=54 (T) Flag=0 Len=24 (4 bytes)
<47_54>
# Row Metadata (24 bytes)
# Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
<47_54_52>
# ROWID: 20 bytes
AAASrPAAEAAAAQ2AAI 1

# Header: Type=5A (Z) Flag=1 Len=202 (4 bytes)
```

The index redo is not extracted into the trail.

The commit redo is not extracted explicitly into the trail.

The total number of bytes required in the GoldenGate trail for the above change is 612.

For array inserts, Oracle redo logging is efficient as all rows inserted together in a single block are grouped together in a single QML operation. Where possible, index insertions are also optimized.

However, GoldenGate does not have an array insert operation. Each array insert is split into separate rows in the GoldenGate trail. Therefore GoldenGate is much less efficient than Oracle redo logging for array inserts.

Direct load inserts

The following statements were executed to prepare the STUDENT table for this example:

```
CREATE TABLE student_temp AS SELECT * FROM student WHERE ROWNUM < 1;
```

```

INSERT INTO student_temp VALUES (1007,'Victoria','Evans','F',
'Oxford','Theology',2013,9000);

INSERT INTO student_temp VALUES (1008,'Katy','Pierce','F',
'Oxford','Theology',2013,9000);

INSERT INTO student_temp VALUES (1009,'Shane','Thomas','M',
'Manchester','Media Studies',2013,9000);

DELETE FROM student WHERE student_key IN (1007,1008,1009);

COMMIT;

```

A number of conditions must be satisfied in order for a direct load insert to be used. One of these is that the target table must not have any referential constraints. The STUDENT table has a referential constraint against the COURSE table which must be disabled for this test:

```

SQL> ALTER TABLE student DISABLE CONSTRAINT student_course;

Table altered.

```

Consider the following statements:

```

INSERT /*+ APPEND */ INTO student SELECT * FROM student_temp;

COMMIT;

```

The above INSERT statement generates the following entries in the redo log.

The first redo record allocates a slot in an undo segment header for the transaction.

```

REDO RECORD - Thread:1 RBA: 0x00004d.00000005.0010 LEN: 0x0140 VLD: 0x05
SCN: 0x0000.00193819 SUBSCN: 1 04/01/2013 18:04:53
(LWN RBA: 0x00004d.00000005.0010 LEN: 0001 NST: 0001 SCN: 0x0000.00193819)
CHANGE #1 TYP:0 CLS:23 AFN:3 DBA:0x00c000b0 OBJ:4294967295 SCN:0x0000.001937f0
SEQ:1 OP:5.2 ENC:0 RBL:0
ktudh redo: slt: 0x0015 sqn: 0x00000361 flg: 0x0011 siz: 80 fbi: 0
uba: 0x00c000b7.0201.23 pxid: 0x0000.000.00000000
CHANGE #2 TYP:0 CLS:24 AFN:3 DBA:0x00c000b7 OBJ:4294967295 SCN:0x0000.001937ef
SEQ:2 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 80 spc: 2582 flg: 0x0010 seq: 0x0201 rec: 0x23
xid: 0x0004.015.00000361
ktubl redo: slt: 21 rci: 0 opc: 5.7 [objn: 0 objd: 0 tsn: 0]
Undo type: Regular undo Begin trans Last buffer split: No
Temp Object: No
Tablespace Undo: No
0x00000000 prev ctl uba: 0x00c000b7.0201.21
prev ctl max cmt scn: 0x0000.00192ab1 prev tx cmt scn: 0x0000.00192ac5
txn start scn: 0xffff.ffffffff logon user: 90 prev brb: 12583093 prev bcl: 0
BuExt idx: 0 flg2: 0
CHANGE #3 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:5.20 ENC:0
session number = 48
serial number = 661
transaction name =
version 186647296
audit sessionid 220372
Client Id =
login username = US03

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	68
1	5.2	Allocate transaction slot in undo header	60

2	5.1	Undo for transaction start	128
3	5.20	Media recovery marker	64

In this case there is no redo. Apparently all necessary information is stored in the undo record.

The next few redo records create a recursive transaction to allocate a new block from the segment bitmap (ASSM). These operations are out of scope for this post.

The next redo record inserts an array of rows into the STUDENT table using a direct load operation:

```

REDO RECORD - Thread:1 RBA: 0x00004d.0000000b.00b4 LEN: 0x2050 VLD: 0x01
SCN: 0x0000.00193820 SUBSCN: 1 04/01/2013 18:04:53

CHANGE #1 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:24.6 ENC:0

CHANGE #2 TYP:1 CLS:1 AFN:4 DBA:0x01000460 OBJ:76495 SCN:0x0000.00193820 SEQ:1
OP:19.1 ENC:0 RBL:0
Direct Loader block redo entry
Block header dump: 0x0bbf0000
Object id on Block? Y
seg/obj: 0x12acf csc: 0x00.19381c itc: 3 flg: E typ: 1 - DATA
brn: 1 bdba: 0x1000430 ver: 0x01 opc: 0
inc: 0 exflg: 0

  Itl          Xid          Uba          Flag  Lck          Scn/Fsc
0x01    0x0004.015.00000361  0x00000000.0000.00  ----    0  fsc 0x0000.00000000
0x02    0x0000.000.00000000  0x00000000.0000.00  ----    0  fsc 0x0000.00000000
0x03    0x0000.000.00000000  0x00000000.0000.00  ----    0  fsc 0x0000.00000000
bdba: 0x0bbf0000
data_block_dump,data header at 0x7f59f4523088
=====
tsiz: 0x1f80
hsiz: 0x18

pbl: 0x7f59f4523088
     76543210
flag=-----
ntab=1
nrow=3
frre=-1
fsbo=0x18
fseo=0x1eef
avsp=0x1ed7
tosp=0x1ed7
0xe:pti[0]      nrow=3  offs=0
0x12:pri[0]      offs=0x1f51
0x14:pri[1]      offs=0x1f25
0x16:pri[2]      offs=0x1eef
block_row_dump:
tab 0, row 0, @0x1f51
tl: 47 fb: --H-FL-- lb: 0x0  cc: 8
col 0: [ 3]  c2 0b 08
col 1: [ 8]  56 69 63 74 6f 72 69 61
col 2: [ 5]  45 76 61 6e 73
col 3: [ 1]  46
col 4: [ 6]  4f 78 66 6f 72 64
col 5: [ 8]  54 68 65 6f 6c 6f 67 79
col 6: [ 3]  c2 15 0e
col 7: [ 2]  c2 5b
tab 0, row 1, @0x1f25
tl: 44 fb: --H-FL-- lb: 0x0  cc: 8
col 0: [ 3]  c2 0b 09
col 1: [ 4]  4b 61 74 79
col 2: [ 6]  50 69 65 72 63 65
col 3: [ 1]  46
col 4: [ 6]  4f 78 66 6f 72 64

```

```

col 5: [ 8] 54 68 65 6f 6c 6f 67 79
col 6: [ 3] c2 15 0e
col 7: [ 2] c2 5b
tab 0, row 2, @0xleef
tl: 54 fb: --H-FL-- lb: 0x0 cc: 8
col 0: [ 3] c2 0b 0a
col 1: [ 5] 53 68 61 6e 65
col 2: [ 6] 54 68 6f 6d 61 73
col 3: [ 1] 4d
col 5: [13] 4d 65 64 69 61 20 53 74 75 64 69 65 73
col 6: [ 3] c2 15 0e
col 7: [ 2] c2 5b
end_of_block_dump
Dump of memory from 0x00007F59F4525008 to 0x00007F59F4525009
7F59F4525000 00000006 [....]

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	24.6	Media recovery marker	44
2	19.1	Direct load block	8204

The size of the second change is a function of the database block size.

The next redo record inserts an array of index rows into the STUDENT_PK index:

```

REDO RECORD - Thread:1 RBA: 0x00004d.0000001c.005c LEN: 0x01a8 VLD: 0x01
SCN: 0x0000.00193820 SUBSCN: 1 04/01/2013 18:04:53
CHANGE #1 TYP:0 CLS:23 AFN:3 DBA:0x00c000b0 OBJ:4294967295 SCN:0x0000.0019381f
SEQ:1 OP:5.2 ENC:0 RBL:0
ktudh redo: slt: 0x0015 sqn: 0x00000000 flg: 0x0002 siz: 144 fbi: 255
uba: 0x00c000b7.0201.24 pxid: 0x0000.000.00000000
CHANGE #2 TYP:0 CLS:24 AFN:3 DBA:0x00c000b7 OBJ:4294967295 SCN:0x0000.00193819
SEQ:1 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 144 spc: 2500 flg: 0x0022 seq: 0x0201 rec: 0x24
xid: 0x0004.015.00000361
ktubu redo: slt: 21 rci: 0 opc: 10.22 objn: 76492 objd: 76493 tsn: 4
Undo type: Regular undo Undo type: Last buffer split: No
Tablespace Undo: No
0x00000000
index undo for leaf key operations
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0008.00c.0000040e uba: 0x00c01a19.0103.09
flg: C--- lkc: 0 scn: 0x0000.001937e7
Dump kdilk : itl=2, kdxlklflg=0x21 sdc=-931123752 indexid=0x1000342 block=0x01000343
(kdxlpu): purge leaf row
number of keys: 3
key sizes:
12 13 15
key : (40):
04 4b 61 74 79 06 50 69 65 72 63 65 05 53 68 61 6e 65 06 54 68 6f 6d 61 73
08 56 69 63 74 6f 72 69 61 05 45 76 61 6e 73
keydata/bitmap: (1): ff
CHANGE #3 TYP:0 CLS:1 AFN:4 DBA:0x01000343 OBJ:76493 SCN:0x0000.00193820 SEQ:1
OP:10.2 ENC:0 RBL:0
index redo (kdxlin): insert leaf row
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0004.015.00000361 uba: 0x00c000b7.0201.24
REDO: ARRAY / -- / --
itl: 2, sno: 1, row size 70

```

```

number of keys: 3
slots:
 1 5 9
insert key: (40):
 04 4b 61 74 79 06 50 69 65 72 63 65 05 53 68 61 6e 65 06 54 68 6f 6d 61 73
 08 56 69 63 74 6f 72 69 61 05 45 76 61 6e 73
each key size is:
 12 13 15
keydata: (18): 01 00 04 60 00 01 01 00 04 60 00 02 01 00 04 60 00 00

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.2	Dummy undo header allocation	60
2	5.1	Undo for insert index rows	192
3	10.2	Redo for insert index rows	148

In this example, all the index rows can be stored on the index root block. Therefore only one block is affected by the change. If more than one block was affected by the above change, then separate redo records would be required for each block modified.

The next redo record inserts an array of index entries into the STUDENT_UK index:

```

REDO RECORD - Thread:1 RBA: 0x00004d.0000001d.005c LEN: 0x0134 VLD: 0x01
SCN: 0x0000.00193820 SUBSCN: 1 04/01/2013 18:04:53

CHANGE #1 TYP:0 CLS:24 AFN:3 DBA:0x00c000b7 OBJ:4294967295 SCN:0x0000.00193820
SEQ:1 OP:5.1 ENC:0 RBL:0
ktudb redo: siz: 116 spc: 2354 flg: 0x0022 seq: 0x0201 rec: 0x25
          xid: 0x0004.015.00000361
ktubu redo: slt: 21 rci: 36 opc: 10.22 objn: 76491 objd: 76494 tsn: 4
Undo type: Regular undo          Undo type: Last buffer split: No
Tablespace Undo: No
          0x00000000
index undo for leaf key operations
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0008.00c.0000040e uba: 0x00c01a19.0103.0a
          flg: C--- lkc: 0          scn: 0x0000.001937e7
Dump kdilk : itl=2, kdxlkflg=0x21 sdc=-931123752 indexid=0x100043a block=0x0100043b
(kdxlpu): purge leaf row
number of keys: 3
key sizes:
 4 4 4
key : (12): 03 c2 0b 08 03 c2 0b 09 03 c2 0b 0a
keydata/bitmap: (1): ff
CHANGE #2 TYP:0 CLS:1 AFN:4 DBA:0x0100043b OBJ:76494 SCN:0x0000.00193820 SEQ:1
OP:10.2 ENC:0 RBL:0
index redo (kdxlin): insert leaf row
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0004.015.00000361          uba: 0x00c000b7.0201.25
REDO: ARRAY / -- / --
itl: 2, sno: 5, row size 42
number of keys: 3
slots:
 5 6 7
insert key: (12): 03 c2 0b 08 03 c2 0b 09 03 c2 0b 0a
each key size is:
 4 4 4
keydata: (18): 01 00 04 60 00 00 01 00 04 60 00 01 01 00 04 60 00 02

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
2	5.1	Undo for insert index rows	164
3	10.2	Redo for insert index rows	120

Again all the index rows can be stored on the index root block. Therefore only one block is affected by the change.

The COMMIT statement generates the following redo:

```

REDO RECORD - Thread:1 RBA: 0x00004d.00000020.0014 LEN: 0x00c0 VLD: 0x01
SCN: 0x0000.00193821 SUBSCN: 1 04/01/2013 18:04:53
CHANGE #1 TYP:0 CLS:23 AFN:3 DBA:0x00c000b0 OBJ:4294967295 SCN:0x0000.00193820
SEQ:2 OP:5.4 ENC:0 RBL:0
ktucm redo: slt: 0x0015 sqn: 0x00000361 srt: 0 sta: 9 flg: 0x12 ktucf redo: uba:
0x00c000b7.0201.28 ext: 0 spc: 1898 fbi: 1
CHANGE #2 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ:0 OP:24.4 ENC:0

```

The above redo record consists of the following components:

Change #	Operation	Description	Bytes
		Redo record header	24
1	5.4	Release transaction slot in undo header	100
2	24.4	Media recovery marker	68

Note that several redo records have been omitted from the above discussion as they relate to segment allocation and management.

The redo log entries generate the following TrailAnalyzer output from the GoldenGate extract trail:

The following is added to the GoldenGate trail for the first row inserted.

```

# Header: Type=47 (G) Flag=1 Len=218 (4 bytes)

# Row (210 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
  # Row Header (47 bytes)
  # Flags - 0x05 Insert
  # Timestamp 2013:04:01 18:04:53 000000 ms
  # Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=104 (4 bytes)
<47_44>
  # Row Data (104 bytes)
  # Slot 0 (8 bytes)
  00 00 00 04 31 30 30 37
  # Slot 1 (12 bytes)
  00 00 00 08 56 69 63 74 6F 72 69 61
  # Slot 2 (9 bytes)
  00 00 00 05 45 76 61 6E 73
  # Slot 3 (5 bytes)
  00 00 00 01 46
  # Slot 4 (10 bytes)
  00 00 00 06 4F 78 66 6F 72 64
  # Slot 5 (12 bytes)
  00 00 00 08 54 68 65 6F 6C 6F 67 79
  # Slot 6 (8 bytes)
  00 00 00 04 32 30 31 33
  # Slot 7 (8 bytes)

```

```

00 00 00 04 39 30 30 30

# Header: Type=54 (T) Flag=0 Len=47 (4 bytes)
<47_54>
# Row Metadata (47 bytes)
# Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
<47_54_52>
# ROWID: 20 bytes
AAASrPAAEAAAARgAAA 1

# Header: Type=4C (L) Flag=0 Len=7 (4 bytes)
<47_54_4C>
# Commit SCN (7 bytes)
1652769

# Header: Type=36 (6) Flag=0 Len=8 (4 bytes)
<47_54_36>
# Transaction ID (8 bytes)
4.21.865

# Header: Type=5A (Z) Flag=1 Len=218 (4 bytes)

```

The following is added to the GoldenGate trail for the second row inserted.

```

# Header: Type=47 (G) Flag=1 Len=192 (4 bytes)

# Row (184 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
# Row Header (47 bytes)
# Flags - 0x05 Insert
# Timestamp 2013:04:01 18:04:53 000000 ms
# Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=101 (4 bytes)
<47_44>
# Row Data (101 bytes)
# Slot 0 (8 bytes)
00 00 00 04 31 30 30 38
# Slot 1 (8 bytes)
00 00 00 04 4B 61 74 79
# Slot 2 (10 bytes)
00 00 00 06 50 69 65 72 63 65
# Slot 3 (5 bytes)
00 00 00 01 46
# Slot 4 (10 bytes)
00 00 00 06 4F 78 66 6F 72 64
# Slot 5 (12 bytes)
00 00 00 08 54 68 65 6F 6C 6F 67 79
# Slot 6 (8 bytes)
00 00 00 04 32 30 31 33
# Slot 7 (8 bytes)
00 00 00 04 39 30 30 30

# Header: Type=54 (T) Flag=0 Len=24 (4 bytes)
<47_54>
# Row Metadata (24 bytes)
# Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
<47_54_52>
# ROWID: 20 bytes
AAASrPAAEAAAARgAAB 1

# Header: Type=5A (Z) Flag=1 Len=192 (4 bytes)

```

The following is added to the GoldenGate trail for the third row inserted.


```

# Header: Type=47 (G) Flag=1 Len=202 (4 bytes)

# Row (194 bytes)
# Header: Type=48 (H) Flag=0 Len=47 (4 bytes)
<47_48>
# Row Header (47 bytes)
# Flags - 0x05 Insert
# Timestamp 2013:04:01 18:04:53 000000 ms
# Object US03.STUDENT

# Header: Type=44 (D) Flag=0 Len=111 (4 bytes)
<47_44>
# Row Data (111 bytes)
# Slot 0 (8 bytes)
00 00 00 04 31 30 30 39
# Slot 1 (9 bytes)
00 00 00 05 53 68 61 6E 65
# Slot 2 (10 bytes)
00 00 00 06 54 68 6F 6D 61 73
# Slot 3 (5 bytes)
00 00 00 01 4D
# Slot 4 (14 bytes)
00 00 00 0A 4D 61 6E 63 68 65 73 74 65 72
# Slot 5 (17 bytes)
00 00 00 0D 4D 65 64 69 61 20 53 74 75 64 69 65
73
# Slot 6 (8 bytes)
00 00 00 04 32 30 31 33
# Slot 7 (8 bytes)
00 00 00 04 39 30 30 30

# Header: Type=54 (T) Flag=0 Len=24 (4 bytes)
<47_54>
# Row Metadata (24 bytes)
# Header: Type=52 (R) Flag=0 Len=20 (4 bytes)
<47_54_52>
# ROWID: 20 bytes
AAASrPAAEAAAARgAAC 1

# Header: Type=5A (Z) Flag=1 Len=202 (4 bytes)

```

The index redo is not extracted into the trail.

The commit redo is not extracted explicitly into the trail.

The total number of bytes required in the GoldenGate trail for the above change is 612. This is the same as required for the array insert.

For direct loads, Oracle redo logging is efficient when blocks contain a large number of rows. This is because the redo overhead per row is lower. Direct loads are faster in other areas of the kernel as they bypass several layers of the kernel.

For small numbers of rows Oracle redo logging less efficient because of the overheads associated with segment allocation and management.

However, GoldenGate does not have a direct load operation. Therefore each direct load block is split into separate rows in the GoldenGate trail. Therefore GoldenGate is much less efficient than Oracle redo logging for direct loads. In fact for insertion of a deterministic set of rows, GoldenGate appears to handle both array inserts and direct loads identically.

After performing the test described above, the constraint should be re-enabled using the following command:

```

SQL> ALTER TABLE student ENABLE CONSTRAINT student_course;

Table altered.

```

