

# Recovering Object Data using Flashback

---

Flashback technologies use data from the Undo tablespace to recover data.

How far back you can go to retrieve data is a fairly inexact science and is based on ...

- undo\_retention parameter (Value in seconds)
- How busy the database is and the size of the Undo tablespace

On all BU live databases the undo\_retention is set to 28880 which is 8 hours.

No flashback operations can take place if the structure of the table has been changed between the current time and the flashback time.

Using flashback you can perform the following actions

- View data using a SQL query from previous date/time using **Flashback Query**
- View the changes to a table during a specified date/time using **Flashback Version**
- Recover a table to a previous point in time using **Flashback Table**
- Recover a dropped table using **Flashback Drop**
- Retrieve all undo for a transaction using **Flashback Transaction Query**
- Automatically back out a transaction using **Flashback Transaction Backout**

The below is an example of amending data on UNTETEST.

For Flashback Query, Version and Transaction see ...

[http://docs.oracle.com/cd/E11882\\_01/appdev.112/e25518/adfns\\_flashback.htm#g1026131](http://docs.oracle.com/cd/E11882_01/appdev.112/e25518/adfns_flashback.htm#g1026131)

For Flashback Table and Drop see ...

[http://docs.oracle.com/cd/E11882\\_01/backup.112/e10642/rcmflash.htm#i1018669](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/rcmflash.htm#i1018669)

## Flashback Query

Show query at exact time

```
SELECT * FROM CAPD_PERSON
AS OF TIMESTAMP TO_TIMESTAMP('12-JUN-13 14:35:00','DD-MON-YY HH24:MI:SS');
```

Show query from exactly 1 hour ago

```
SELECT * FROM CAPD_PERSON
AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '1' HOUR);
```

Use DBMS\_FLASHBACK function instead of AS OF TIMESTAMP

```
EXECUTE DBMS_FLASHBACK.ENABLE_AT_TIME(TO_TIMESTAMP('12-JUN-13 14:35:00','DD-MON-YY
HH24:MI:SS'));
SELECT * FROM CAPD_PERSON;
EXECUTE DBMS_FLASHBACK.DISABLE;
```

## **Flashback Version**

Specify exact time (May receive ORA-30052 error if not enough undo)

```
SELECT versions_startscn, versions_starttime, versions_endscn,
versions_endtime, versions_xid, versions_operation,
p_forenames, p_surname, p_dob
FROM capd_person VERSIONS BETWEEN TIMESTAMP
TO_TIMESTAMP('12-JUN-13 09:40:00', 'DD-MON-YY HH24:MI:SS') AND
TO_TIMESTAMP('12-JUN-13 10:40:00', 'DD-MON-YY HH24:MI:SS')
WHERE VERSIONS_OPERATION is not null
ORDER BY p_surname desc, VERSIONS_STARTSCN
```

Specify all possible flashback data for this table

```
SELECT versions_startscn, versions_starttime, versions_endscn,
versions_endtime, versions_xid, versions_operation,
p_forenames, p_surname, p_dob
FROM capd_person VERSIONS BETWEEN TIMESTAMP MINVALUE AND MAXVALUE
WHERE VERSIONS_OPERATION is not null
ORDER BY p_surname desc, VERSIONS_STARTSCN
```

## **Flashback Table**

You may have to enable row movement before performing the flashback operation

```
ALTER TABLE CAPD_PERSON ENABLE ROW MOVEMENT;
```

Flashback single table

```
FLASHBACK TABLE person TO TIMESTAMP TO_TIMESTAMP('21-MAY-10 14:35:00','DD-MON-YY
HH24:MI:SS');
```

Flashback more than one table

```
FLASHBACK TABLE person, person_detail TO TIMESTAMP TO_TIMESTAMP('21-MAY-10
14:35:00','DD-MON-YY HH24:MI:SS');
```

You can also flashback to a restore point or a SCN.

## **Flashback Drop**

This command will recover dropped tables from the recycle bin.

In SQL\*Plus use the command SHOW RECYCLEBIN to see if the table is there and confirm the drop time.

```
FLASHBACK TABLE DUBELIVE.BU_STAFF TO BEFORE DROP;
```

You can also assign a new name to the restored table

```
FLASHBACK TABLE DUBELIVE.BU_STAFF TO BEFORE DROP RENAME TO DUBELIVE.BU_STAFF2;
```

After restoring the tables, any indexes will still have recyclebin names (i.e. begin with BIN\$).

Also the recyclebin does not preserve referential constraints on the table.

## Flashback Transaction Query

Will retrieve all the undo for a particular transaction.

First discover the transaction ID, then query the FLASHBACK\_TRANSACTION\_QUERY view using this ID to display the undo needed to recover from this transaction.

```
IAN@RMAN11G> select * from football;
```

ID	NAME	LEAGUE
1	Ipswich Town	Championship
2	Southampton	Premiership

```
IAN@RMAN11G> delete from football;
```

```
2 rows deleted.
```

```
IAN@RMAN11G> commit;
```

```
Commit complete.
```

```
IAN@RMAN11G> SELECT NAME, VERSIONS_XID FROM FOOTBALL VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE;
```

NAME	VERSIONS_XID
Southampton	03001100CD3F0000
Ipswich Town	03001100CD3F0000
Ipswich Town	
Southampton	

```
IAN@RMAN11G> conn / as sysdba
```

```
Connected.
```

```
SYS@RMAN11G> SELECT OPERATION,UNDO_SQL FROM FLASHBACK_TRANSACTION_QUERY  
WHERE XID=HEXTORAW('03001100CD3F0000');
```

OPERATION	UNDO_SQL
DELETE	insert into "IAN"."FOOTBALL"("ID","NAME","LEAGUE") values ('2', 'Southampton', 'Premiership');
DELETE	insert into "IAN"."FOOTBALL"("ID","NAME","LEAGUE") values ('1', 'Ipswich Town', 'Championship');

## Flashback Transaction Backout

To use this functionality the database must have at least minimal supplemental logging enabled.

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```

This is much easier to perform using Cloud Control.

### Select **Availability – Backup & Recovery – Transactions**

This will take you to the LogMiner screen which is fully integrated with Flashback Transaction.

You can now search for your transaction based on a time range or filter on a table or schema name.

Click on the Transaction ID of the transaction you wish to flashback and select **Flashback Transaction**

Unfortunately testing shows that even a simple transaction (such as the example in Flashback Transaction Query) will need supplemental logging on the Primary Key. We currently do not have this enabled on any databases. This option can double the size of log files and cause transactions to take slightly longer to complete.