New and Improved Methods for Administering Your Database

December 16, 2004

Howard Horowitz

Objective

Expose you to some of the database features available in 10g and compare them to the lengthy workarounds that were used in previous Oracle versions.

10g provides better ways to administer your database. *"It takes the mountain out of the mole hill".*

Mountains & Mole Hills

- **9i Mountains = More effort and resources to accomplish similar tasks in 10g**

- **10g Mole Hills = Less effort and resources to accomplish similar tasks in 9i.**

Popular Features

- **Automatic Shared Memory Management (ASMM)**
- *Data Pump*
- *SQL Tuning Advisor*
- *Flashback Database*
- *RMAN - Backupset Compression*

Automatic Shared Memory Management (ASMM)

8i method for automating SGA management

There is no method.

W**orkaround**

You have to shutdown the database and manually change the values. This could be done programmatically with multiple init<SID>.ora files. Each file containing different values for the SGA parameters and automated via shell and Cron/Autosys.

Automatic Shared Memory Management (ASMM)

9i method for automating SGA management.

Still not doable, however, you can dynamically change many of the values without shutting down the database.

Workaround

You have to use the alter system/session commands and also rely on the v$shared_pool_advice and db_cache_advice views for proper settings. Manual / programmatic effort is required if the behavior of your database changes and SGA changes are needed. Cron and Autosys to automate.

Automatic Shared Memory Management (ASMM)

10g method for automating SGA management.

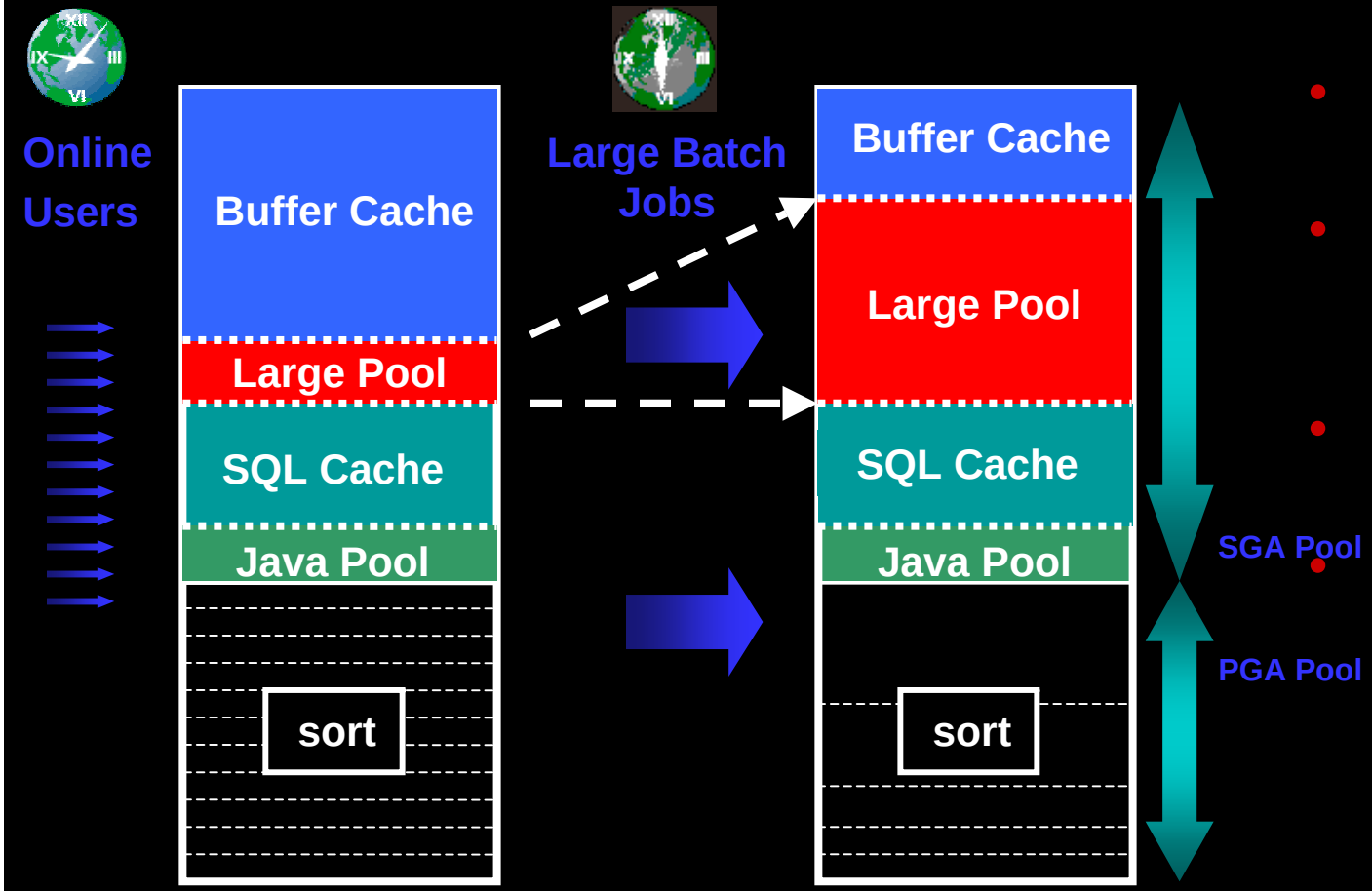
alter system set sga_target='x';

Automatic Shared Memory Management (ASMM)

sga_target -- This parameter is new in Oracle Database 10*g* and reflects the total size of memory an SGA can consume.

- Shared pool
- Buffer cache
- Java Pool
- Large Pool

Automatic Shared Memory Management (ASMM)



- Automatically adapts to workload changes
- Maximizes memory utilization
- Single Parameter makes it easier to use
- Helps eliminate out of memory errors
- Can help improve performance

Automatic Shared Memory Management (ASMM)

- Requires an SPFILE and SGA_TARGET > 0. Can not exceed sga_max_size.
- Does not apply to the following parameters.
  - Log Buffer
  - Other Buffer Caches (KEEP/RECYCLE, other block sizes)
  - Streams Pool (new in Oracle Database 10g)
  - Fixed SGA and other internal allocations
- Can be adjusted via EM or command line.
- A new background process named Memory Manager (MMAN) manages the automatic shared memory.

Automatic Shared Memory Management

DEMO

Popular Features

- *Automatic Shared Memory Management (ASMM)*
- **Data Pump**
- *SQL Tuning Advisor*
- *Flashback Database*
- *RMAN - Backupset Compression*

Data Pump

- 8i / 9i method for suspending exports and imports. **N/A**
- 8i / 9i method for restarting failed exports and imports at point of failure. **N/A**
- 8i / 9i method for controlling the number of threads/processes. **N/A**
- 8i / 9i method for direct mode imports. **N/A**
- 8i / 9i method for monitoring export and import's. **N/A**
- 8i / 9i method for importing and exporting data via PL/SQL. **N/A**
- 8i / 9i method for exporting/importing pre-defined objects via include or exclude keywords (grants, procedures, functions, tables..etc). Supports like and not like clause. **N/A**
- 8i / 9i method for remapping tablespaces and datafiles. **N/A**

Data Pump

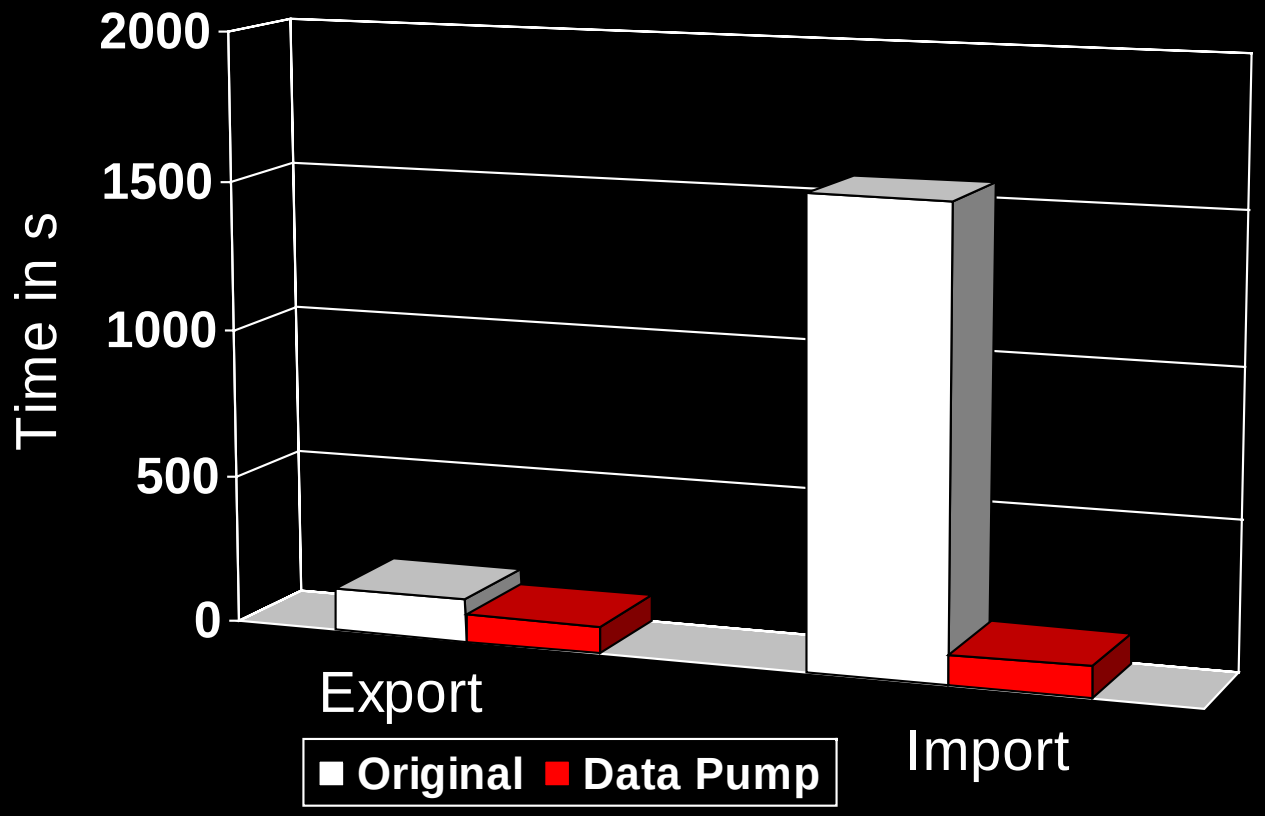
High performance import and export

- 60% faster than 9i export (single thread)
- 15x-45x faster than 9i import (single thread)

The reason it is so much faster is that Conventional Import uses only conventional mode inserts, whereas Data Pump Import uses the Direct Path method of loading. As with Export, the job can be parallelized for even more improvement dynamically. Creates a separate dump file for each degree of parallelism.

Data Pump

Time is money. Data Pump has cut down data movement/processing times significantly.



Popular Features

- *Automatic Shared Memory Management (ASMM)*
- *Data Pump*
- **SQL Tuning Advisor**
- *Flashback Database*
- *RMAN - Backupset Compression*

SQL Tuning Advisor

Most database and application related problems are the result of poorly written SQL and missing or misused indexes.

Oracle provides an interface through OEM or via the PL/SQL stored packages dbms_advisor and dbms_sqltune to analyze existing SQL statements, provide tuning recommendations and implement those recommendations.

- http://servername.com:5501/em/
- Navigate to performance
- Then to Advisor Central
- SQL Tuning Advisor
- Top SQL

Additional advisors: redo log size advisor, sql access advisor, undo advisor, and segment advisor.

Popular Features

- *Automatic Shared Memory Management (ASMM)*
- *Data Pump*
- *SQL Tuning Advisor*
- **Flashback Database**
- *RMAN - Backupset Compression*

Flashback Database

8i / 9i method for point-in-time recovery

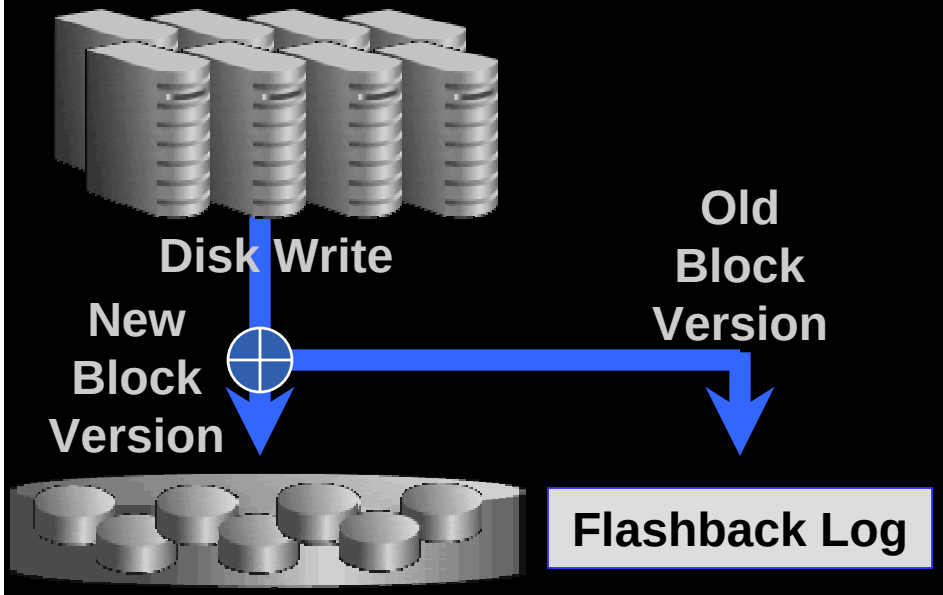
- Shutdown the database
- **Restore all of the datafiles from last backup**
- Startup the database in mount state
- Recover database until (SCN or Time)
- **Apply the necessary redo/archive logs**
- Open the database – open resetlogs

Flashback Database

10g method for point-in-time recovery

- Shutdown the database
- Startup the database in mount state
- SQL> ***flashback database to timestamp to_timestamp('2004-12-16 16:10:00', 'YYYY-MM-DD HH24:MI:SS');***
- Open the database – open resetlogs

Flashback Database



Like a "Rewind" button for the Database

- New strategy for point-in-time recovery
- Flashback Log captures old versions of changed blocks.
  - Think of it as a continuous backup
  - Replay log to restore DB to time
  - Restores just changed blocks
- It's fast - recovers in minutes, not hours. More over, this feature removes the need for database incomplete recoveries that require physical movement of datafiles/restores.
- It's easy - single command restore
  - *SQL> Flashback Database to scn 1329643*

Flashback Database

Restrictions

- Not used for Media failure errors. Used for Logical/User errors.
- The database control file has been restored or re-created.
- Previous tablespace has been dropped.
- The database data file that contains the object to be queried has been shrunk.
- A recovery through the resetlogs command has occurred.

Views for Monitoring

- V$Database
- V$Flashback_Database_Log
- V$Flashback_Database_Stat

Flashback Database

DEMO

Popular Features

- *Automatic Shared Memory Management (ASMM)*
- *Data Pump*
- *SQL Tuning Advisor*
- *Flashback Database*
- **RMAN - Backupset Compression**

RMAN – Backupset Compression

8i / 9i method for compressing backups
(Compression utility)

gzip *.bak, *.arc, *.ctl….etc;

RMAN – Backupset Compression

10g method for compressing backups

- RMAN> *CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO COMPRESSED BACKUPSET;*
- *RMAN> BACKUP AS COMPRESSED BACKUPSET DATABASE PLUS ARCHIVELOG;*

Prior to Oracle 10g, RMAN reduced the size of backup images by backing up only used blocks. This was great for databases that were over-sized, however, this didn't help for large databases with little free space.

The AS COMPRESSED BACKUPSET option of the BACKUP command allows RMAN to perform binary compression of backupsets. The resulting backupsets do not need to be uncompressed during recovery.

RMAN – Backupset Compression

Pros:

- Backupsets were compressed by 78% when compared to a regular backupset.

Cons:

- Creating compressed backupsets imposes some extra CPU overhead during backup and restore, which can slow the backup process. If you have more than one CPU, you can use increased parallelism to run jobs on multiple CPUs and thus improve performance.

Hidden Gems

- **Renaming Tablespaces**
- *Dictionary View Improvements*
- *Flush Buffer Cache*
- *Automated Statistics Collection*
- *Flashback Drop*
- *Flashback Table*
- *Flashback Transaction Query*
- *Diamonds in the rough*

Rename Tablespace

8i / 9i method for renaming tablespaces

- Create a new tablespace with the same size as the original one. (You have to make sure you have enough room on disk to store a duplicate copy). Space pending, this might require additional analysis of the original tablespace to determine if the new tablespace can be resized/reorged.
- Move objects from the original tablespace to the new one. (This could take a while, depending on the size of the tablespace).
- Drop the original tablespace and datafile(s) after the objects are moved to the newly named tablespace.

Rename Tablespace

10g method for renaming tablespaces

*SQL> alter tablespace users **rename** to users3;*

Rename Tablespace

Oracle allows the renaming of tablespaces in 10g. A simple alter tablespace command is all you need.

*SQL> alter tablespace users **rename** to users3;*

Tablespace altered.

Elapsed: 00:00:00.05

SQL> *alter tablespace users3 **rename** to users;*

Tablespace altered.

Elapsed: 00:00:00.02

Rename Tablespace

- Rename tablespace feature has lessened the workload for TTS operations. There's no need to delete tablespaces on the target prior to impdp metadata.

- Doesn't Support System or Sysaux tablespaces

- Supports Default, Temporary, and Undo Tablespaces (dynamically changes the spfile).

Rename Tablespace

DEMO

Hidden Gems

- *Renaming Tablespaces*
- **Dictionary View Improvements**
- *Flush Buffer Cache*
- *Automated Statistics Collection*
- *Flashback Drop*
- *Flashback Table*
- *Flashback Transaction Query*
- *Diamonds in the rough*

Dictionary View Improvements

8i / 9i method for monitoring blocking locks

UTLLOCKT.sql – requires catblock.sql to be run

OR

--Blocking Locks Info

```
SELECT
 bs.username "Blocking User",
 bs.username "DB User",
 ws.username "Waiting User",
 bs.sid "SID",
 ws.sid "WSID",
 bs.sql_address "address",
 bs.sql_hash_value "Sql hash",
 bs.program "Blocking App",
 ws.program "Waiting App",
 bs.machine "Blocking Machine",
 ws.machine "Waiting Machine",
 bs.osuser "Blocking OS User",
 ws.osuser "Waiting OS User",
 bs.serial# "Serial#",
 DECODE(wk.TYPE,
 'MR', 'Media Recovery',   'RT', 'Redo Thread',  'UN', 'USER Name',
 'TX', 'Transaction',  'TM', 'DML', 'UL', 'PL/SQL USER LOCK',
 'DX', 'Distributed Xaction', 'CF', 'Control FILE', 'IS', 'Instance State',
 'FS', 'FILE SET', 'IR', 'Instance Recovery', 'ST', 'Disk SPACE Transaction',
……..
FROM
   v$lock hk,  v$session bs,
……..
……..
```

Dictionary View Improvements

10g method for monitoring blocking locks

New columns in v$session allow you to easily identify sessions that are blocking and waiting for other sessions. V$session also contains information from v$session_wait view. No need to join the two views.

--Display blocked session and their blocking session details.

```
SELECT  sid, serial#, blocking_session_status, blocking_session
FROM     v$session
WHERE   blocking_session IS NOT NULL;
```

or

```
SELECT  blocking_session_status, blocking_session
FROM    v$session
WHERE sid = 444;  /* Blocked Session */
```

Dictionary View Improvements

8i/9i method for monitoring rollback activity

In addition to monitoring how long a SQL session will take, now you can monitor rollback activity from one dictionary view. Prior to 10g, you had to do the following:

Select b.sid, a.used_ublk

From v$transaction a, v$session b

Where a.addr=b.taddr and b.username = 'user-name'

V$session.used_ublk (Used Undo Block) will give you the count of number of blocks to be rolled back. Take counts every five minutes to figure out how long it will take to rollback a transaction. This is a manual process, leaving room for error.

Dictionary View Improvements

10g method for monitoring rollback activity

```
Select  time_remaining
from    v$session_longops
where   sid = 444; /* rollback session */
```

Or

```
select   time_remaining,sofar,elapsed_seconds
from     v$session_longops l, v$session s
where    l.sid=s.sid and l.serial# = s.serial#
and      s.module='long_proc';
```

SIDE NOTE:

You can also use dbms_appliation_info.set_module and set_client_info to track the progress of a procedure and query the results from v$session_longops. I used to use these packages to monitor procedures via v$session.

Hidden Gems

- *Renaming Tablespaces*
- *Dictionary View Improvements*
- **Flush Buffer Cache**
- *Automated Statistics Collection*
- *Flashback Drop*
- *Flashback Table*
- *Flashback Transaction Query*
- *Diamonds in the rough*

Flush Buffer Cache

8i/9i method for flushing the buffer cache

Prior to 10g, this wasn't possible without shutting down and restarting the database or using the following undocumented commands:

- SQL> *alter session set events = 'immediate trace name flush_cache';*
- alter tablespace offline/online to flush the buffer cache of blocks relating to that tablespace (As per Tom Kytes Article).

Side-Note - You were able to flush the shared pool

SQL> ALTER SYSTEM FLUSH SHARED_POOL;

Flush Buffer Cache

10g method for flushing the buffer cache

10g has provided the ability to flush the buffer cache. This isn't suggested for a production environment, but might be useful for QA/Testing. The bigger the cache, the larger the LRU and dirty list becomes. That results in longer search times. However, if the buffer cache is undersized, than running the following command can improve performance and take the burden off the DBWR. In addition to decreasing free buffer waits.

SQL> ALTER SYSTEM FLUSH BUFFER_CACHE;

Flush Buffer Cache

DEMO

Hidden Gems

- *Renaming Tablespaces*
- *Dictionary View Improvements*
- *Flush Buffer Cache*
- **Automated Statistics Collection**
- *Flashback Drop*
- *Flashback Table*
- *Flashback Transaction Query*
- *Diamonds in the rough*

Automatic Statistics Collection

8i/9i method for gathering statistics

Required a scheduled process that called DBMS_STATS or DBMS_UTIL packages. For finer granularity, master-slave scripts may have been created that called dbms_stats/analyze commands based on the percentage of table/index changes.

```
for sid in test1 test2 test3
do
echo "connecting to $sid"
sqlplus /nolog << EOF
SET ECHO ON
SET SERVEROUT ON SIZE 1000000
CONNECT system/manager@$sid


BEGIN
FOR uname IN
( SELECT
username FROM dba_users
WHERE username NOT IN ('SYS', 'SYSTEM', 'OUTLN', 'DBSNMP')
)
LOOP
    DBMS_OUTPUT.PUT_LINE (
    'Analyzing USER :'|| uname.username);

    DBMS_STATS.GATHER_SCHEMA_STATS(
    uname.username, estimate_percent => '25', block_sample => TRUE,
    method_opt => 'FOR ALL INDEXED COLUMNS SIZE 1',
    degree => '2', granularity => 'ALL', cascade => TRUE,
    options => 'GATHER');
END LOOP;
END;
/
exit
EOF
done
```

Automatic Statistics Collection

10g method for gathering statistics

Database statistics are automatically collected using the *dbms_stats.gather_database_stats_job_proc procedure.*

- By default this job runs within a maintenance window between 10 P.M. to 6 A.M. week nights and all day on weekends.
- *DBMS_STATS.GATHER_DATABASE_STATS_JOB_PROC* is an internal procedure which gathers statistics for tables with either empty or stale statistics, similar to the DBMS_STATS.GATHER_DATABASE_STATS procedure using the GATHER AUTO option. The main difference is that the internal job prioritizes the work such that tables most urgently requiring statistics updates are processed first.

Automatic Statistics Collection

You can also prevent statistics from being overwritten via dbms_stats.lock_schema_stats and unlock. In 10g you can also restore statistics to any point in time, in case the new statistics that were collected cause a sub optimal plan to be generated. You can restore statistics for a table, schema, fixed database objects, or the entire database.

Hidden Gems

- *Renaming Tablespaces*
- *Dictionary View Improvements*
- *Flush Buffer Cache*
- *Automated Statistics Collection*
- **Flashback Drop**
- *Flashback Table*
- *Flashback Transaction Query*
- *Diamonds in the rough*

Flashback Drop

8i/9i method for undoing a dropped table

- Table level import from logical backup
- Restore backup to another location and export table from backup database. Import dropped table from backup database to original database.
- Point-in-time recovery prior to the dropped table

The first method is the fastest but will require a fair amount of time, depending on the table size. The second method is an extension of the first if a logical backup wasn't taken. The third method requires a database shutdown.

Flashback Drop

10g method for undoing a dropped table

SQL> *flashback table emp to before drop;*

Flashback Drop

- Recycle Bin (Sounds familiar)…….A logical representation of the dropped object. The dropped/renamed table is still occupying space in it's original tablespace. You can still query it after it's dropped.

- You can empty out the recycle bin by purging the objects.

  select object_name, original_name, type
  from user_recycle_bin; or show recycle;

  Purge table mm$$55777$table$1;

- You can permanently drop a table without writing to recycle bin.

  Drop table emp purge;

Flashback Drop

Has a few quirks

- Doesn’t restore foreign key constraints or materialized views.
- Restores indexes, constraints, and triggers with it’s klingon language - *mm$$55777$table$1* (Requires a rename of triggers and constraints).

Flashback Drop

DEMO

Hidden Gems

- *Renaming Tablespaces*
- *Dictionary View Improvements*
- *Flush Buffer Cache*
- *Automated Statistics Collection*
- *Flashback Drop*
- **Flashback Table**
- *Flashback Transaction Query*
- *Diamonds in the rough*

Flashback Table

8i method for restoring data

- Point-in-time recovery from backupset
- Logical import from export
- Restore database to new location and import or direct path insert into source table via dblink.

Flashback Table

9i method for restoring data

```
INSERT INTO emp
  (SELECT *
   FROM emp AS OF TIMESTAMP
  TO_TIMESTAMP('16-Sep-04
  1:00:00','DD-MON-YY HH24:MI:SS')
  MINUS
  SELECT * FROM emp
 );
```

Side-Note – This might not work if DDL operations are performed or constraints are altered.

Flashback Table

10g method for restoring data

Flashback table emp to TIMESTAMP TO_TIMESTAMP('16-Sep-04 1:00:00','DD-MON-YY HH24:MI:SS')

Make sure you enable row movement prior to the restore.
SQL> alter table emp enable row movement;

Side-Note – This might not work if DDL operations are performed or constraints are altered.

Flashback Table

The before image data is stored in your undo_tablespace. Make sure you allocate enough space. Also make sure you allocate a big retention_size if you decide not to let Oracle automatically manage it. Setting undo_retention=0 tells oracle to automatically manage this parameter.

I'm off on a tangent again, but it's a good one.

Oracle will prevent you from getting those annoying ora-01555 snapshot too old errors by including *"retention guarantee"* in your create or alter statements. This does come at a cost if your running a transaction that needs undo space. You can disable it *"alter tablespace retention noguarantee;"*

Flashback Table

DEMO

Hidden Gems

- *Renaming Tablespaces*
- *Dictionary View Improvements*
- *Flush Buffer Cache*
- *Automated Statistics Collection*
- *Flashback Drop*
- *Flashback Table*
- **Flashback Transaction Query**
- *Diamonds in the rough*

Flashback Transaction Query

8i/9i method for generating sql undo statements

Log Miner (Good luck parsing through those logs).

Flashback Transaction Query

10g method for generating sql undo statements

```
SELECT undo_sql
FROM   flashback_transaction_query
WHERE  table_owner='SCOTT'
AND    table_name='EMP'
AND    start_scn between 21553 and 44933;
```

(You can also use timestamp)

Flashback Transaction Query provides the ability to generate the SQL statements for undoing DML .

Hidden Gems

- *Renaming Tablespaces*
- *Dictionary View Improvements*
- *Flush Buffer Cache*
- *Automated Statistics Collection*
- *Flashback Drop*
- *Flashback Table*
- *Flashback Transaction Query*
- **Diamonds in the rough**

Diamonds in the rough

Additional 10g features worth mentioning

- Drop database command (includes datafiles, control files, archive logs, backups, and spfile).

  RMAN> drop database including backups;

- Freeing up unused space in tables/indexes and adjusting the high-water mark. Also requires *"alter table emp enable row movement"*

  SQL> alter table emp shrink space cascade
  compact;

- utl_mail (no need to reference utl_smtp protocol. It's built in)
- utl_compress (compression of binary data (blobs and raw data). Similar to gzip.
- Support of regular expressions (Unix commands in PL/SQL)
- Default temporary and user tablespaces

Putting it all together

*10g has made major strides in preventing DBAs and developers from making **mountains** out of **mole hills**. These gains have led to more efficient code and better database administration. The underlying result is savings in cost and time for your organization.*

References

Web Sites:

http://www.oracle.com/technology/pub/articles/10gdba/index.html
(Oracle Database 10g: Top 20 Features for DBAs

http://www.adpgmbh.ch/ora/misc/dynamic_performance_views.html

Books:

Oracle Database 10g New Features

Presentations:

Tuning Oracle 10g, Rich Niemiec

Howard.Horowitz@adeccona.com
hhorow6801@aol.com