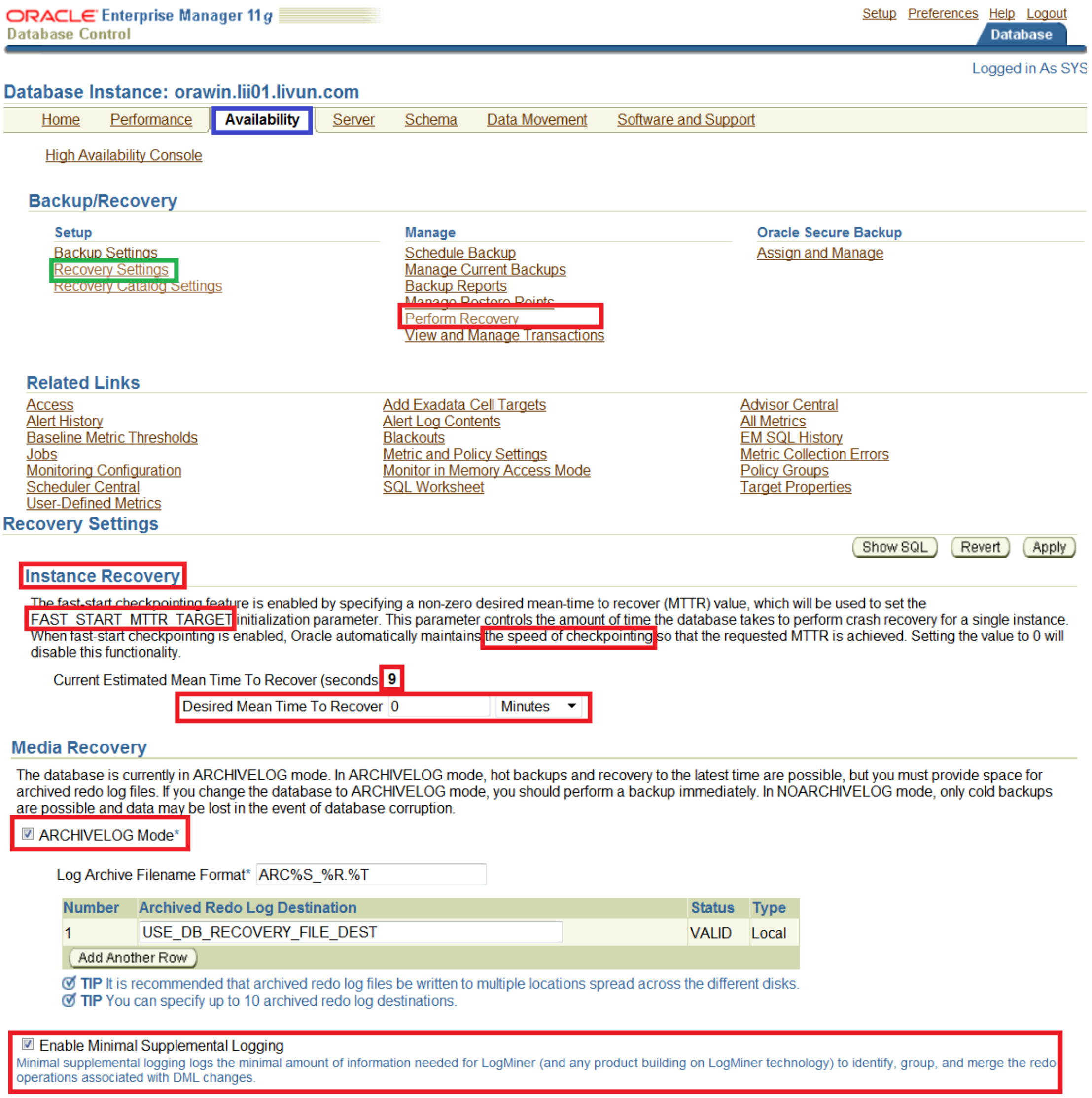
Data Flash Back

Human error has always been the most difficult type of error from which to recover. This is because as far as the database is concerned, human error is not an error at all. The “error” is just another committed transaction, and the rules of a relational database (the “D” for “Durable” of the ACID test) do not allow Oracle to back out committed transactions. Depending on the nature of the error, the different Flashback technologies may help you to recover, while minimizing downtime and loss of data.



Database Instance: orawin.lii01.livun.com

Recovery Settings

Instance Recovery

The fast-start checkpointing feature is enabled by specifying a non-zero desired mean-time to recover (MTTR) value, which will be used to set the FAST_START_MTTR_TARGET initialization parameter. This parameter controls the amount of time the database takes to perform crash recovery for a single instance. When fast-start checkpointing is enabled, Oracle automatically maintains the speed of checkpointing so that the requested MTTR is achieved. Setting the value to 0 will disable this functionality.

Current Estimated Mean Time To Recover (seconds **9**

Desired Mean Time To Recover 0 Minutes ▾

Media Recovery

The database is currently in ARCHIVELOG mode. In ARCHIVELOG mode, hot backups and recovery to the latest time are possible, but you must provide space for archived redo log files. If you change the database to ARCHIVELOG mode, you should perform a backup immediately. In NOARCHIVELOG mode, only cold backups are possible and data may be lost in the event of database corruption.

☑ ARCHIVELOG Mode*

Log Archive Filename Format* ARC%S_%R.%T

Number	Archived Redo Log Destination	Status	Type
1	USE_DB_RECOVERY_FILE_DEST	VALID	Local

✓ **TIP** It is recommended that archived redo log files be written to multiple locations spread across the different disks.
✓ **TIP** You can specify up to 10 archived redo log destinations.

☑ Enable Minimal Supplemental Logging
Minimal supplemental logging logs the minimal amount of information needed for LogMiner (and any product building on LogMiner technology) to identify, group, and merge the redo operations associated with DML changes.

The typical reason for using any type of flashback technology is to correct mistakes.There are four distinct flashback technologies available, each implemented with a different underlying architecture. Each technology has different capabilities and limitations, but there is overlapping functionality between them. The typical reason for using any type of flashback technology is to correct mistakes—it is vital to understand what type of flashback technology is appropriate for correcting different types of errors.

Flashback Database: Flashback Database is, by analogy, like pressing a rewind button on the database. The current database is taken as the starting point, and it is taken back in time, change by change, reversing all work done sequentially. The end result is as if you had done an incomplete recovery: all work subsequent to the flashback point is lost, and indeed the database must be opened with RESETLOGS.

Flashback Database is an alternative to incomplete recovery following user error—perhaps a more flexible and much faster alternative. Flashback Database is in fact a combination of several processes and data structures. First, you must allocate some memory in the SGA (which will be automatic - you cannot control how large the buffer is) and some space on disk to store the flashback data, and start the RVWR process to enable flashback logging. When doing a flashback, Oracle will use the flashback logs to take the database back in time to before the time you want, and then apply redo logs (using whatever archive redo log files and online redo log files are necessary) in the usual fashion for incomplete recovery to bring the datafiles forward to the exact time you want. Then the database can be opened with a new incarnation, in the same manner as following a normal incomplete recovery.

Flashback Database is restricted by the size of the `flashback logs.` Unlike the redo log, flashback logging is not a log of changes—it is a log of complete block images.

```
SQL> alter system set db_recovery_file_dest='/flash_recovery_area';
SQL> alter system set db_recovery_file_dest_size=8G;
SQL> alter system set db_flashback_retention_target=240;
SQL> shutdown immediate;
SQL> startup mount;
SQL> alter database open;
```

The flash recovery area is a default location for all recovery-related files: either permanent or transient. The permanent files are:

- Controlfile multiplexed copies
- Online redo log file multiplexed copies

The transient occupants of the flash recovery area are

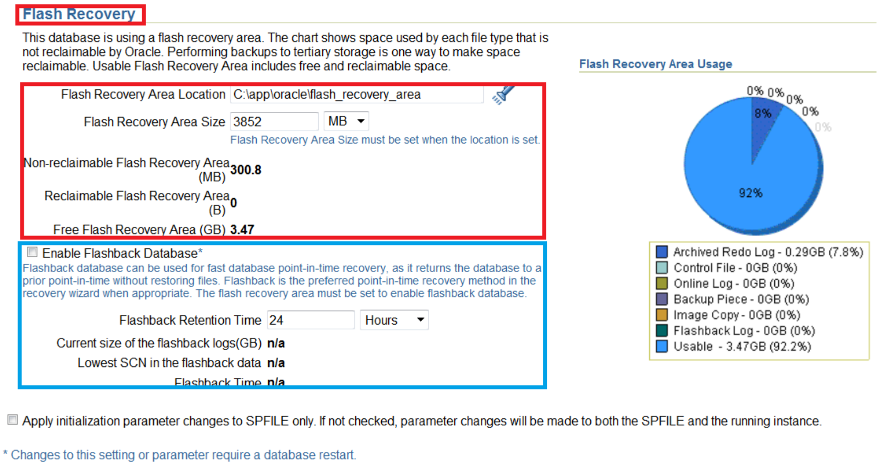
- Archive redo log files
- RMAN backups
- RMAN copies
- Controlfile auto-backups
- Flashback log files

The flash recovery area is controlled by two instance parameters:

- DB_RECOVERY_FILE_DEST
- DB_RECOVERY_FILE_DEST_SIZE

```
SQL> show parameters recovery
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
db_recovery_file_dest            string      C:\app\oracle\recovery_area
db_recovery_file_dest_size      big integer 3852M
```

The first of these parameters nominates the location. The second parameter limits the maximum amount of space in the destination that the database will take up, Oracle Corporation’s advice is that the flash recovery area should be greater than the sum of the sizes of the database, all incremental backups since the last full backup, and all archive redo log files that have not been backed up to tape.



Flash Recovery

This database is using a flash recovery area. The chart shows space used by each file type that is not reclaimable by Oracle. Performing backups to tertiary storage is one way to make space reclaimable. Usable Flash Recovery Area includes free and reclaimable space.

Flash Recovery Area Location	C:\app\oracle\flash_recovery_area
Flash Recovery Area Size	3852 MB
	Flash Recovery Area Size must be set when the location is set.
Non-reclaimable Flash Recovery Area (MB)	**300.8**
Reclaimable Flash Recovery Area (B)	**0**
Free Flash Recovery Area (GB)	**3.47**

□ Enable Flashback Database*

Flashback database can be used for fast database point-in-time recovery, as it returns the database to a prior point-in-time without restoring files. Flashback is the preferred point-in-time recovery method in the recovery wizard when appropriate. The flash recovery area must be set to enable flashback database.

Flashback Retention Time	24 Hours
Current size of the flashback logs(GB)	**n/a**
Lowest SCN in the flashback data	**n/a**
Flashback Time	**n/a**

Flash Recovery Area Usage

- Archived Redo Log - 0.29GB (7.8%)
- Control File - 0GB (0%)
- Online Log - 0GB (0%)
- Backup Piece - 0GB (0%)
- Image Copy - 0GB (0%)
- Flashback Log - 0GB (0%)
- Usable - 3.47GB (92.2%)

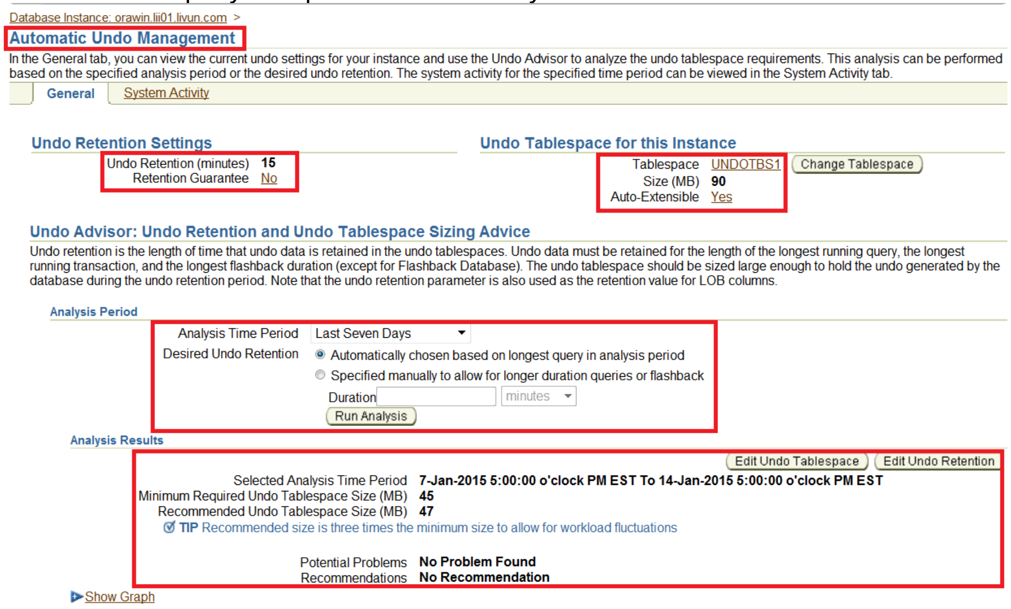
□ Apply initialization parameter changes to SPFILE only. If not checked, parameter changes will be made to both the SPFILE and the running instance.

* Changes to this setting or parameter require a database restart.

Flashback Query, Transaction, and Table: three flashback techniques are based on the use of undo segments. The first flashback capability was initially introduced with release 9i of the database and has been substantially enhanced subsequently. Flashback Query (the release 9i feature) lets you query the database as it was at some time in the past, either for one select statement or by taking your session temporarily back in time so that all its queries will be against a previous version of the database. Flashback Transaction automates the repair process. Once you have used Flashback Query to identify which transaction it was that caused the problem, Oracle can construct SQL statements that will reverse the changes. Having determined that inappropriate work has been committed against one table, you can instruct Oracle to reverse all changes made to that table since a particular point in time, while leaving all other tables current.

Throughout any Flashback Query, Transaction, or Table operation, the database remains open and all objects (including those involved in the flashback) are available for use. Transactional integrity and constraints are always enforced, which means that the flashback operation might fail. For example, if a flashback of a transaction requires an insert into a primary key column, that value must not be in use. Flashing back one table may not be possible if it has foreign key constraints—you will have to flash back all the related tables in one operation.

The flashback query is capable and limited by the undo retention:



Automatic Undo Management

In the General tab, you can view the current undo settings for your instance and use the Undo Advisor to analyze the undo tablespace requirements. This analysis can be performed based on the specified analysis period or the desired undo retention. The system activity for the specified time period can be viewed in the System Activity tab.

Undo Retention Settings

Undo Retention (minutes)	**15**
Retention Guarantee	No

Undo Tablespace for this Instance

Tablespace	UNDOTBS1
Size (MB)	**90**
Auto-Extensible	Yes

Undo Advisor: Undo Retention and Undo Tablespace Sizing Advice

Undo retention is the length of time that undo data is retained in the undo tablespaces. Undo data must be retained for the length of the longest running query, the longest running transaction, and the longest flashback duration (except for Flashback Database). The undo tablespace should be sized large enough to hold the undo generated by the database during the undo retention period. Note that the undo retention parameter is also used as the retention value for LOB columns.

Analysis Period

Analysis Time Period: Last Seven Days

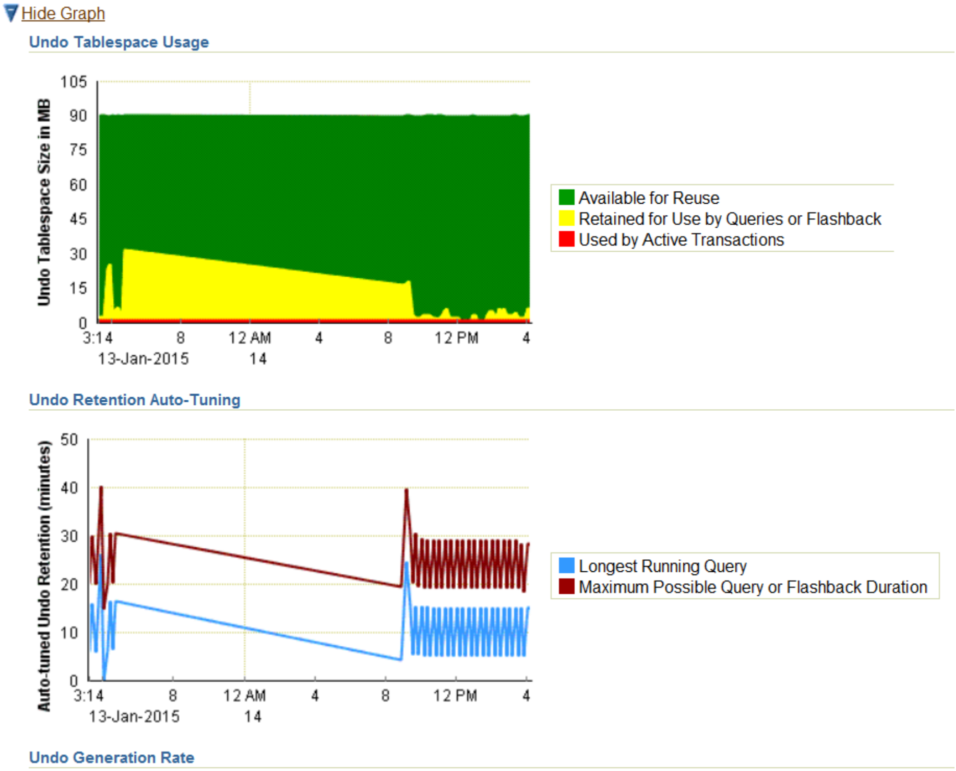
Desired Undo Retention:
● Automatically chosen based on longest query in analysis period
○ Specified manually to allow for longer duration queries or flashback
Duration ____ minutes

Analysis Results

Selected Analysis Time Period	**7-Jan-2015 5:00:00 o'clock PM EST To 14-Jan-2015 5:00:00 o'clock PM EST**
Minimum Required Undo Tablespace Size (MB)	**45**
Recommended Undo Tablespace Size (MB)	**47**

✓ TIP Recommended size is three times the minimum size to allow for workload fluctuations

Potential Problems	**No Problem Found**
Recommendations	**No Recommendation**



Undo Tablespace Usage

Undo Tablespace Size in MB
105
90
75
60
45
30
15
0
3:14
8
12 AM
4
8
12 PM
4
13-Jan-2015
14
🟩 Available for Reuse
🟨 Retained for Use by Queries or Flashback
🟥 Used by Active Transactions

Undo Retention Auto-Tuning

Auto-tuned Undo Retention (minutes)
50
40
30
20
10
0
3:14
8
12 AM
4
8
12 PM
4
13-Jan-2015
14
🟦 Longest Running Query
🟥 Maximum Possible Query or Flashback Duration

Undo Generation Rate

To guarantee that a flashback query will always succeed for a given period, set the `RETENTION GUARANTEE` attribute for the undo tablespace, in conjunction with the UNDO_RETENTION instance parameter. This will ensure that you can always flash back the number of seconds specified—but the price you will pay is that if your undo tablespace is not sized adequately for the transaction workload, then the database may hang while performing DML, you must monitor the `V$UNDOSTAT` view to calculate the necessary size.

Flashback Drop, It is now possible to “un-drop” a table. This is implemented by mapping the DROP command onto a RENAME command. Rather than dropping the table, it is renamed to a system-generated name, and only actually dropped later, when its storage space is needed for a live object. If necessary, and if its storage space has not been reused, the object can be renamed back to its original name and thus restored. Without this capability, the only way to get a table back after a drop would be to do an incomplete recovery to the point in time just before the table was dropped. This was usually timeconsuming, and it meant the loss of all work done subsequently. The new Flashback Database capability achieves the same result as incomplete recovery and should be much faster, but work done on other tables following the drop is lost and the database will be unavailable until the operation is completed.

```
SQL> drop table <table_name> purge;
SQL> flashback table <table_name> to before drop rename to <new_name>;
```

Flashback Drop lets you reinstate the table as it was at the time that it was dropped, with no loss of data whatsoever; the database remains current. This does not require any use of backups, and neither is there any downtime for users. Note that Flashback Drop is specifically for the DROP command; you cannot flash back a TRUNCATE command. Along with the table itself, any associated indexes and permissions will also be restored. You cannot flash back a table truncation, only a table drop. Database Control also has an interface to Flashback Drop. From the database home page, take the Availability tab then Perform Recovery in the Manage section. In the drop-down box for Recovery Scope, select TABLES, and then the radio button for Flashback Dropped Tables. `The recycle bin` is a term given to the storage space used by `dropped objects`.

The capacity of flashback drop is limited by the available space in tablespaces

Enabling a table for Flashback Data Archive creates another table (and a few other objects) that will store all previous versions of rows: storing them forever if desired. When DML is committed against the table, a background process called the Flashback Data Archive process (the FBDA) will capture the necessary data and save it to the archive. From there, it can be queried with the same syntax used for a regular flashback query - but the flashback can go back years.

The Flashback Data Archive is for long-term storage. Typically, this will be for legal reasons: in many jurisdictions there are requirements for keeping data for years, and then for destroying it (sometimes known as “digital shredding”). A Flashback Data Archive can enable this transparently. When a Flashback Data Archive is created, you specify a time limit (which may be years) and the FBDA will make sure that all data is saved until that time has passed, and will then remove it. The FBDA is responsible for creating the objects in the archive, populating them with rows as necessary and purging data that has passed the expiry date.

There are three data dictionary views that document the Flashback Data Archive configuration:

- DBA_FLASHBACK_ARCHIVE Describes the configured archives.
- DBA_FLASHBACK_ARCHIVE_TS Shows the quotas assigned per archive per tablespace.
- DBA_FLASHBACK_ARCHIVE_TABLES Lists the tables for which archiving are enabled.