

# BACKUP & RECOVERY

# CHECKPOINT

# CHECKPOINT

---

## ❖ Checkpointing

- Saves information about the current status of, and changes to, the database to data files
- Management tasks after checkpointing has completed
  - Change the current recovery start point to the end point of checkpointing
  - Delete unneeded redo log files that have been applied to data files
- Regular checkpointing reduces the amount of time required to recover the database in the event of a fault.

## ❖ ALTIBASE HDB Checkpointing features

- Fuzzy & Ping-pong checkpointing
  - Transaction performance is not reduced during checkpointing
  - Two sets of data files are maintained
  - Only applies to memory tablespaces

# CHECKPOINT

---

## ❖ Performing Checkpointing

### ➤ Automatic execution (Default)

- Set auto-execution using the CHECKPOINT\_ENABLED property
  - ◆ Checkpointing interval: - CHECKPOINT\_INTERVAL\_IN\_SEC
  - ◆ Number of redo log files- CHECKPOINT\_INTERVAL\_IN\_LOG

### ➤ Manual execution

- User command - Only the SYS user can perform checkpointing manually.

```
iSQL> ALTER SYSTEM CHECKPOINT;
```

## ❖ Reasons why redo log files might not be deleted after checkpointing is completed:

- They might contain information about transactions that are still underway
- In Archivelog mode, they have not yet been archived
- If using replication, they have not yet been converted to XLogs and sent for replication

# BACKUP

# DATABASE MODE

## ❖ Database mode

- Redo log files are managed in one of two different ways after checkpointing:
  - Archivelog mode – Archived to a specified path and then deleted
  - Noarchivelog mode – Deleted without being archived
- Set when the CREATE DATABASE statement is executed
  - Can be subsequently changed in the CONTROL startup phase (restart required)

## ❖ Comparison

Database Mode	Pros	Cons
Archivelog mode	<ul style="list-style-type: none"><li>▪ Media recovery possible</li><li>▪ If archive log files are not lost, recovery up to the present time is possible.</li></ul>	<ul style="list-style-type: none"><li>▪ Disk space for saving archive log files must be provided.</li><li>▪ DBA is burdened with additional archive log file management tasks</li></ul>
Noarchivelog mode	<ul style="list-style-type: none"><li>▪ No disk space for saving archive log files is needed.</li><li>▪ DBA need not perform archive log file management tasks</li></ul>	<ul style="list-style-type: none"><li>▪ Media recovery is impossible.</li><li>▪ Recovery is only possible up to the point at which the most recent offline backup was conducted.</li></ul>

# DATABASE MODE

---

## ❖ Checking the current database mode

- More detailed information can be checked by querying the v\$archive performance view

```
iSQL(sysdba)> SELECT archive_mode, archive_dest FROM v$archive;  
ARCHIVE_MODE      ARCHIVE_DEST  
-----  
0                /edu/alt11/altibase_home/arch_logs
```

## ❖ Changing the database mode after the database is created

- Start up ALTIBASE HDB in the CONTROL phase
- Change the database mode
  - Can only be changed by the SYS user in SYSDBA mode

```
iSQL> connect sys/manager as sysdba  
iSQL(sysdba)> shutdown immediate;  
iSQL(sysdba)> startup control;  
iSQL(sysdba)> ALTER DATABASE ARCHIVELOG;  
iSQL(sysdba)> startup;
```

# DATABASE BACKUP

## ❖ Types of database backup and objects that are backed up

Backup Type	Target Object	Recovery Point	DB Mode
Online Database Backup	The entire database (data files, log anchor files) except redo log files		
Online Tablespace Backup	The data files for a particular tablespace	▪ Present time point	▪ Must be Archivelog mode
Online Tablespace backup by “file system copy”			
Online log anchor backup			
Offline backup	The entire database (data files, log anchor files, redo log files)	▪ Time of last backup	▪ Noarchivelog or Archivelog mode
Backup using aexport and iloader	Schema creation script, data in a particular table		

- Online backup-related commands can only be executed by the “SYS” user, in SYSDBA mode
  - Cannot be executed from multiple sessions, as there can be only one SYSDBA mode session at any time.



# DATABASE BACKUP

---

## ❖ Online Database Backup

- Back up the entire database at particular time points while ALTIBASE HDB is running
- When the command is executed, all data files and redo loganchor files are automatically backed up to the specified directory
  - Only one of two (#0 and #1) checkpoint image files is backed up
  - Temporary tablespace data files are not backed up, as they are only used for processing queries

```
iSQL(sysdba)> ALTER DATABASE BACKUP DATABASE TO 'backup_dir';
```

# DATABASE BACKUP

---

## ❖ Online Tablespace Backup

- Back up a particular tablespace while ALTIBASE HDB is running
- The data files related to the tablespace are automatically backed up to the specified path when the command is executed.
  - Only one of two (#0 and #1) memory tablespace-related data files is backed up

```
iSQL(sysdba)> ALTER DATABASE BACKUP  
TABLESPACE tablespace_name TO 'backup_dir';
```

# DATABASE BACKUP

---

## ❖ Online Tablespace Backup by DBA

- Backup particular tablespace while DB is operating
- Copy tablespace datafile using OS copy command after the backup using backup command
- Log switching is essential to forcibly archive the related redo log file
- Interoperable with 3<sup>rd</sup> party backup solution as the multiple tablespaces can be backed up simultaneously

1. iSQL(sysdba)> **ALTER TABLESPACE *tablespace\_name* BEGIN BACKUP;**
2. Copy tablespace datafile using OS copy command
3. iSQL(sysdba)> **ALTER TABLESPACE *tablespace\_name* END BACKUP;**
4. iSQL(sysdba)> **ALTER SYSTEM SWITCH LOGFILE;**

# DATABASE BACKUP

---

## ❖ Online log anchor file backup

- Back up log anchor files while ALTIBASE HDB is running
- All three log anchor files are automatically backed up to the specified path when the query is executed

```
iSQL(sysdba)> ALTER DATABASE BACKUP LOGANCHOR TO 'backup_dir';
```

# DATABASE BACKUP

---

## ❖ Offline database backup

- Back up the entire database after ALTIBASE HDB is shut down.
- Back up data files, redo log files and log anchor files using an OS copy command.

1. iSQL(sysdba)> **shutdown immediate;**
2. Backup data files, redo log files and loganchor files by using OS copy command.
3. iSQL(sysdba)> **startup;**

# DATABASE BACKUP

---

## ❖ Example of offline database backup

### ➤ altibase.properties settings

```
MEM_DB_DIR = ?/mem_dbs  
DEFAULT_DISK_DB_DIR = ?/dbs  
LOG_DIR = ?/logs  
LOGANCHOR_DIR = ?/logs
```

### ➤ Backup procedure

- This example assumes that the tablespace data files are located in the default directory

```
iSQL(sysdba)> shutdown immediate;  
shell> cp -r $ALTIBASE_HOME/mem_dbs/* /backup  
shell> cp -r $ALTIBASE_HOME/dbs/* /backup  
shell> cp -r $ALTIBASE_HOME/logs /backup  
iSQL(sysdba)> startup;
```

# DATABASE BACKUP

---

## ❖ Backup Considerations

- When performing offline backup, there are more things to check other than the default directory in the properties file
  - Data files in other directories might be missed
  - Check data file-related performance views for data files in other directories.
- Online backup and checkpointing cannot be performed simultaneously
  - If checkpointing is underway when the online backup procedure starts, the backup procedure waits for checkpointing to finish
  - If online backup is underway when checkpointing is attempted, an error occurs
    - [ERR-110A4 : Backup is in progress. Please wait until the current backup process completes.]
- Do one of the followings to prepare for recovery when modifying a tablespace:
  - Back up the tablespace, the dictionary tablespace, and log anchor files
  - Perform online or offline database backup
- If any replication objects exist, replication information is also backed up.
  - If the database is restored on a system with a different IP address, the replication error will occur.

# RECOVERY



# DATABASE RECOVERY

---

## ❖ Restart Recovery

- Automatic recovery in the event of an abnormal shutdown, such as a power outage or a system crash.

```
.....  
TRANSITION TO PHASE : META  
[SM] Recovery Phase - 1 : Preparing Database  
      : Dynamic Memory Version => Parallel Loading  
[SM] Recovery Phase - 2 : Loading Database  
[SM] Recovery Phase - 3 : Starting Recovery  
      Initializing Active Transaction List  
      Redo  
      Refine Disk Table..  
      Undo  
[SM] Refine Memory Table :  
..... [SUCCESS]
```

# DATABASE RECOVERY

---

## ❖ Media recovery

- Performed when data files are missing or corrupted.
- The database should be manually restored using:
  - archive log files
  - backup data files
  - log anchor files
- Recovery commands must be executed in the CONTROL phase. (Online recovery is not possible.)
- Is possible only if ALTIBASE HDB is operating in ArchiveLog mode.
- Both complete and partial recovery are possible

# MEDIA RECOVERY

---

## ❖ Complete recovery

- Execute the command shown below to restore the database up to the present:

```
iSQL(sysdba)> ALTER DATABASE RECOVER DATABASE;
```

## ❖ Considerations

- The archive log files AND the redo log files generated since the most recent complete database backup must not be corrupted.
- The most recently updated log anchor files are needed.

# MEDIA RECOVERY

---

## ❖ Partial recovery

- Restore the database to a specific point in time by executing the command shown below:

- If redo log files or archive log files are missing or corrupted:

```
iSQL(sysdba)> ALTER DATABASE RECOVER DATABASE UNTIL CANCEL;
```

- To restore the database to a specified point in time:

```
iSQL(sysdba)> ALTER DATABASE RECOVER DATABASE UNTIL TIME '2010-09-10:17:55:00';
```

- Redo log files must be reset after performing partial recovery.
  - This step is mandatory in order to avoid restart recovery when the system is restarted.

```
iSQL(sysdba)> ALTER DATABASE db_name META RESETLOGS;
```

## ❖ Consideration

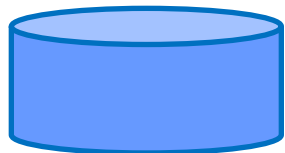
- In order to restore the database to a specific point in time, the log anchor files up to that time are needed.

# MEDIA RECOVERY

## ❖ Recovery example in case of missing data file (1) – recovery using backup data files

- Use the redo log files whose LSN (Log Sequence Number) is between the LSN of the backup data files and the current LSN of the database.

Log Anchor



User1.dbf information  
Creation LSN(20:012284)  
Checkpoint LSN(102:172168)

User1.dbf (backup file)



LSN  
32:345698

Online Redo Logs directory



Log File 101

Log File 102

Recovery command executed



Media Recovery Process

Replay  
Redo logs

Archive Logs Directory



...



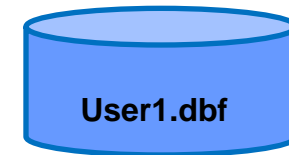
...



Log File 0

Log File 32

Log File 100



LSN  
102:172168

# MEDIA RECOVERY

---

## ❖ Considerations during media recovery

- Some data files must be created by the user.
  - Memory tablespace data files must be copied to form a pair.

```
shell> cp USER_MEM_DATA-1-0 USER_MEM_DATA-0-0;
```

- Empty temporary tablespace data files must be created in the CONTROL phase.

```
iSQL(sysdba)> ALTER DATABASE CREATE DATAFILE 'temp001.dbf';
```

## ❖ Considerations after media recovery

- Back up the database after performing partial recovery.
  - If the database is not backed up at this time, and recovery is again subsequently performed, it will be impossible to recover any changes made to the database after the redo logs were initialized.

# Thank you!

Altibase Education Center

Tel : 02-2082-1451

Fax : 02-2082-1459

E-mail : [education@altibase.com](mailto:education@altibase.com)

Homepage : <http://edu.altibase.com>