

# ■ ■ ■ Backup & Recovery New Features Oracle Database 11g



Peter Jensch

IMS Stuttgart  
29. März 2012

DOAG Regionaltreffen  
Stuttgart

**trivadis**  
makes IT easier. ■ ■ ■

# Agenda – Data Pump



- Data Pump Compressed Dump File Sets
- Expdp new Parameters
- Impdp new Parameters
- Data Pump Legacy Mode



Data are always  
part of the game.

# Agenda - RMAN



Data are always  
part of the game.

- Improved Archivelog Management
- Multi Section Backups
- Improved handling of Archival Backup
- Fast RMAN Compression
- Miscellaneous RMAN Enhancements
- Improved RMAN Scripting
- New RMAN Catalog Features
- RMAN & Data Guard
- Database Duplication
- Data Recovery Advisor

# Compressed Dump File Sets



- Now full compression (data & metadata) of a dump file set is implemented, dumpfile size reduction up to 85 %

```
expdp dp/dp SCHEMAS=hr  
DUMPFILE=DATA_PUMP_DIR:exp_hrcomp01%U.dat, exp_hrcomp02%U.dat  
parallel=2 FILESIZE=250K COMPRESSION=ALL
```

```
COMPRESSION={ALL | DATA_ONLY | METADATA_ONLY | NONE}
```

- **ALL** enables compression for the entire export operation
- **DATA\_ONLY** results in all data being written to the dump file in compressed format
- **METADATA\_ONLY** results in all metadata being written to the dump file in compressed format. This is the default.
- **NONE** disables compression for the entire export operation

# Data Pump – expdp new Parameters



## ■ REMAP\_DATA

- allows transformations to be applied to data during export

```
REMAP_DATA=[schema.]tablename.column_name:[schema.]pkg.function
```

```
expdp scott/tiger \  
  DUMPFILE=expdp_dir:expdp_scott.dmp \  
  LOGFILE=expdp_dir:expdp_scott.log \  
  TABLES=scott.emp,scott.dept \  
  REUSE_DUMPFILLES=yes \  
  REMAP_DATA=scott.emp.sal:scott.pg_chg.change_emp_sal
```

## ■ REUSE\_DUMPFILLES

- Specifies whether or not to overwrite a preexisting dump file

```
expdp scott/tiger DUMPFILE=expdp_dir:expdp_scott.dmp \  
  LOGFILE=expdp_dir:expdp_scott.log \  
  TABLES=scott.emp,scott.dept \  
  REUSE_DUMPFILLES=yes
```

# Data Pump – impdp new Parameters



- **REMAP\_DATA**

- allows transformations to be applied to data during import

```
REMAP_DATA=[schema.]tablename.column_name:[schema.]pkg.function
```

- **REMAP\_TABLE**

- Allows you to rename tables during an import operation

```
impdp scott/tiger DUMPFILE=expdp_dir:expdp_scott.dmp \  
LOGFILE=expdp_dir:impdp.log \  
TABLES=scott.dept REMAP_TABLE=dept:dept_copy \  
EXCLUDE=CONSTRAINT
```

# Data Pump Legacy Mode – Overview



- Backward compatibility for original Export and Import scripts
  - Scripts
  - Parameters
- Possibility to use original Export/Import scripts
  - Migration to Data Pump can be postponed
- Data Pump enters Legacy Mode once it determines an old parameter
- Invoking Data Pump utilities expdp or impdp with original Export/Import syntax necessary
- Original Export and Import utilities are still useful!
- Export/Import scripts migration to Data Pump is still recommended

# Data Pump Legacy Mode – Execution



- Legacy Mode start (original exp utility syntax)

```
expdp userid=system/manager file=sysman_exp.dmp
log=sysman_exp.log compress=no owner=sysman statistics=none
consistent=yes
```

- Automatic translation into Data Pump syntax

```
Connected to: Oracle Database 11g Enterprise Edition Release
11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options
Legacy Mode Active due to the following parameters:
Legacy Mode Parameter: "compress=FALSE" Location: Command
Line, ignored.
Legacy Mode Parameter: "consistent=TRUE" Location: Command
Line, Replaced with: "flashback_time=TO_TIMESTAMP('2009-11-08
18:05:45', 'YYYY-MM-DD HH24:MI:SS')"
```

```
Legacy Mode Parameter: "file=sysman_exp.dmp" Location: Command
Line, Replaced with: "dumpfile=sysman_exp.dmp"
Legacy Mode Parameter: "log=sysman_exp.log" Location: Command
Line, Replaced with: "logfile=sysman_exp.log,,..."
```



# Data Pump Legacy Mode – Directory Objects



- Without specifying FILE parameter the default directory DATA\_PUMP\_DIR will be used

```
oracle@vmoel56ora11g:~/ [ORA11R2] pwd
/u00/app/oracle/product/11.2.0.3/rdbms/log
oracle@vmoel56ora11g:~/product/11.2.0.3/rdbms/log/ [ORA11R2]
ls -ltr sysman*
-rw-r--r-- 1 oracle oinstall      59972 Mar  8 18:11 sysman_exp.log
-rw-r----- 1 oracle oinstall 47120384 Mar  8 18:11 sysman_exp.dmp
```

- With specified FILE parameter in Legacy Mode applicable directory object must exist (Read/Write)!
- Error and crash, if directory object does not exist



# Data Pump Legacy Mode – Exceptions



- Export/Import do not generate log and dump files in the same format as Data Pump
  - Caution by log files monitoring
  - SUCCESSFULLY TERMINATED does not appear in Data Pump log files
  
- Error checking
  - Possible different error messages
  
- Exit status
  - Different exit status values
  
- Detailed Data Pump Export/Import mapping to original Export/Import parameters
  - No suitable mapping → original parameter will be ignored
  - See Oracle Database Utilities 11g Release 2 (11.2)

# Agenda - RMAN



Data are always  
part of the game.

- Improved Archivelog Management
- Multi Section Backups
- Improved handling of Archival Backup
- Fast RMAN Compression
- Miscellaneous RMAN Enhancements
- Improved RMAN Scripting
- New RMAN Catalog Features
- RMAN & Data Guard
- Database Duplication
- Data Recovery Advisor

# Improved Archivelog Management





- Archivelog Failover
  - In the event that the flash recovery area is inaccessible during backup optional archive log destinations can be utilized
- Archivelog Deletion Policy Enhancement

**CONFIGURE ARCHIVELOG DELETION POLICY TO BACKED UP 1  
TIMES TO DISK SHIPPED TO ALL STANDBY**

- the configuration applies to all archiving destinations
- The commands `BACKUP ... DELETE INPUT` and `DELETE ... ARCHIVELOG` obey this configuration, as does the flash recovery area
- additional options: `BACKED UP X TIMES TO DEVICE TYPE, SHIPPED TO [ALL] STANDBY`

# Multi Section Backups (1)



- Multi Section Backup is the possibility to divide the backup of large datafiles into sections.
  - section = contiguous range of blocks in a file
  - all sections have the same size – except the last section
  - a maximum of 256 sections per datafile is possible
  - one backup piece contains one file section
  - all sections of a datafile belong to one backup-set
- This feature improves the performance backups and restores of large datafiles by parallelizing the workload for each file.
- The optimization of aborted backups and restores does not work on section level.

## Multi Section Backups (2)



```
Backup datafile 1 section size 100 M format
'/u00/app/oracle/admin/PKN1/backup/inc0_df1_%d_s%s_p%p';
```

```
Starting backup at 18-MAR-12
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=/u01/oradata/PKN1/system01_PKN1.dbf
backing up blocks 1 through 12800
channel ORA_DISK_1: starting piece 1 at 18-MAR-12
channel ORA_DISK_1: finished piece 1 at 18-MAR-12
piece handle=/u00/app/oracle/admin/PKN1/backup/inc0_df1_PKN1_s7_p1
tag=TAG20070918T171454 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:02
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=/u01/oradata/PKN1/system01_PKN1.dbf
backing up blocks 12801 through 25600
channel ORA_DISK_1: starting piece 2 at 18-MAR-12
channel ORA_DISK_1: finished piece 2 at 18-MAR-12
piece handle=/u00/app/oracle/admin/PKN1/backup/inc0_df1_PKN1_s7_p2
tag=TAG20070918T171454 comment=NONE
```

# Improved handling of Archival Backup (1)



- Archival Backups = Long-term Backups
  - Archival Backups are created with the KEEP Option, which is a self-contained backup.
  - the KEEP option also specifies that the backup should be exempt from the retention policy either forever or for a specified period of time.
  - those backups retain only the archivelogs needed to make the backup consistent.
  
- BACKUP ... KEEP creates automatically:
  - backup of the datafiles, controlfile and spfile
  - backup of the archivelogs, to ensure that the database backup can be recovered to a consistent state
  - a restore point can be specified optionally

**New**

## Improved handling of Archival Backup (2)



- Examples:

```
RMAN> BACKUP DATABASE  
      KEEP FOREVER  
      RESTORE POINT MARCH2012;
```

```
RMAN> BACKUP DATABASE  
      KEEP UNTIL 'SYSDATE+1'  
      RESTORE POINT DUPLTESTDMARCH2012;
```



# Fast RMAN Compression



- In addition to the existing BZIP2 algorithm, RMAN supports the ZLIB algorithm:
  - ZLIB – optimizes time respectively CPU
  - BZIP2 – optimizes maximum compression

```
RMAN> CONFIGURE COMPRESSION ALGORITHM 'ZLIB';
```

- Comparison ZLIB / BZIP2:
  - ZLIB runs faster than BZIP2
  - BZIP2 creates smaller files than ZLIB
  - BZIP2 consumes much more CPU resource
  - ZLIB requires the Oracle Advanced Compression option

# Miscellaneous RMAN Enhancements



- Backup Undo Optimization
  - The backup command does not backup undo that is not needed for recovery of a backup. Undo is not needed for already committed transactions.
  
- Backup of read-only transportable tablespaces
  - RMAN can backup transportable tablespaces when they are not READ/WRITE
  
- Improved block media recovery performance
  - RMAN rather uses blocks from the flashback logs than using full or incremental backups. This can significantly improve block media recovery performance.

# Improved RMAN Scripting



- Substitution variables and USING
  - RMAN scripts can now accept user input at runtime
  - The Substitution of the variables can be done interactively or the values can be specified in a USING clause

```
more test.rcv
    CONNECT TARGET;
    LIST BACKUP OF DATAFILE &1;

rman @test.rcv using 1
RMAN> CONNECT TARGET;
2> LIST BACKUP OF DATAFILE 1;
```

# New RMAN Catalog Features (1)



## ■ IMPORT CATALOG

- IMPORT CATALOG imports the metadata from one recovery catalog schema into a different catalog schema.

```
RMAN> import catalog rman_old/rman_old@catalog_old  
db_name=PROD1 no unregister;
```

## ■ Situations for using IMPORT CATALOG

- to merge one recovery catalog into another recovery catalog
- to switch the recovery catalog database
- to move catalog entries of individual databases

## New RMAN Catalog Features (2)



- Base Catalog / Virtual Private Catalog
  - The owner of a recovery catalog can GRANT or REVOKE access to a subset of the catalog to other database users in the same recovery catalog database.
  - The recovery catalog as a whole is also named base catalog
  - The subset is called a virtual private catalog:  
a virtual private catalog is a set of synonyms and views that refer to a base recovery catalog.
  - This allows the protection of metadata from individual databases

```
RMAN> CONNECT CATALOG rman_vpc/rman_vpc@catalog;  
RMAN> CREATE VIRTUAL CATALOG;  
RMAN> DROP CATALOG;
```

## New RMAN Catalog Features (3)



- Privileges for a virtual private catalog
  - Access privileges can be granted (GRANT) and revoked (REVOKE) from base catalog owner to virtual private catalog owner
  - By default, a virtual private catalog owner has no access to the base recovery catalog
  - GRANT REGISTER allows the virtual private catalog owner to register new target databases in the recovery catalog

```
RMAN> CONNECT CATALOG rman_base/rman_base@catalog;
```

```
RMAN> GRANT CATALOG FOR DATABASE prod1 TO rman_vpc;
```

```
RMAN> GRANT REGISTER DATABASE TO rman_vpc;
```

```
RMAN> REVOKE CATALOG FOR DATABASE prod2 FROM rman_vpc;
```

```
RMAN> REVOKE REGISTER DATABASE FROM rman_vpc;
```

# RMAN & Data Guard (1)



- RMAN uses the initialization parameter `DB_UNIQUE_NAME` to distinguish one database from another
  - Only the primary database must be explicitly registered at the catalog
  - Configure persistent settings for primary and standby databases:

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt FOR DB_UNIQUE_NAME DG11_SITE2;  
CONFIGURE DB_UNIQUE_NAME DG11_SITE2 CONNECT IDENTIFIER  
'DG11_SITE2_connect_string';
```

- Catalog removal of a database in a Data Guard environment

```
UNREGISTER DB_UNIQUE_NAME DG11_SITE1 INCLUDING BACKUPS
```

- Reverse Synchronization: Update from catalog to the controlfile
- `DB_UNIQUE_NAME` enhancement for LIST, REPORT, SHOW, CHANGE

## RMAN & Data Guard (2)



- Explizit resync for databases in a data guard environment:
  - Connect target with **username/password** (!!! ORA-600)
  - Net Service name must be specified for all databases in the FROM DB\_UNIQUE\_NAME Option

```
CONFIGURE DB_UNIQUE_NAME 'DG11_SITE1' CONNECT IDENTIFIER  
'DG11_SITE1' ;
```

- Resync Command

```
RESYNC CATALOG FROM DB_UNIQUE_NAME all;  
RESYNC CATALOG FROM DB_UNIQUE_NAME DG11_SITE2;
```

executes a normal resynchronization and a reverse Synchronization (Persistent Configurations out of the Recovery Catalog will be written into the controlfile)



## RMAN & Data Guard (3)



- Tape Backups are accessible to all databases in the environment
- Disk Backups are accessible only to the database that created them
  - to restore a disk backup from another database in the data guard environment, the disk backup first have to be cataloged or with `CHANGE RESET DB_UNIQUE_NAME` be associated with the other database

```
CONNECT TARGET@PRIMARY;  
CHANGE BACKUP FOR DB_UNIQUE_NAME STANDBY RESET  
DB_UNIQUE_NAME;
```

# RMAN & Data Guard (4)



- Simplified Restore of the controlfiles

```
Thu Feb 02 00:44:37 2012
```

```
Conversion to standby controlfile pending for restored file Wed Oct 03  
00:44:47 2007 alter database mount Converting controlfile to standby If  
db_file_name_convert or log_file_name_convert parameters are not used,  
then RMAN intervention is required to fix the file names in the  
converted control file. Refer to RMAN documentation for how to fix all  
file names.
```

```
Clearing standby activation ID 344023288 (0x148160f8) The primary  
database controlfile was created using the 'MAXLOGFILES 32' clause.  
There is space for up to 29 standby redo logfiles Use the following SQL  
commands on the standby database to create standby redo logfiles that  
match the primary database:
```

```
ALTER DATABASE ADD STANDBY LOGFILE 'sr11.f' SIZE 4194304; ALTER DATABASE  
ADD STANDBY LOGFILE 'sr12.f' SIZE 4194304; ALTER DATABASE ADD STANDBY  
LOGFILE 'sr13.f' SIZE 4194304; ALTER DATABASE ADD STANDBY LOGFILE  
'sr14.f' SIZE 4194304; Set as converted control file due to  
db_unique_name mismatch Changing di2dbun from DG11_SITE1 to DG11_SITE2
```

## RMAN & Data Guard (5)



- Fast incremental backup for physical standby database is included in Active Data Guard option
  - Incremental backups no longer scan the entire database
  - During a backup of a standby database RMAN can use the block change tracking file, to identify the changed blocks since the last incremental backup
  - Technique already known since 10g, but only available for Single Instance & RAC databases
- Recovery after loss of a datafile of the primary database

```
RMAN> CONNECT TARGET SYS@STANDBY
```

```
RMAN> CONNECT AUXILIARY SYS@PRIMARY
```

```
RMAN> BACKUP AS COPY DATAFILE 4 AUXILIARY FORMAT  
    '/u01/oradata/DG11/users01DG11.dbf' ;
```

# Active Database Duplication (1)



- Copy of the target database through the network (backup as copy) => no pre-existing Backup will be restored
  - The auxiliary database must be available through Oracle Net
  - A passwordfile must exist for the auxiliary database
  - The sysdba password of the target and auxiliary database must be the same
  - There is no need to allocate an auxiliary channel (exception: for parallelization reasons)
  - No until clause is possible

## Active Database Duplication (2)



- Example script:

```
CONNECT TARGET sys/manager@prod
CONNECT CATALOG rman/rman@catalog
CONNECT AUXILIARY sys/manager
RUN {
  ALLOCATE CHANNEL ch1 DEVICE TYPE DISK;
  ALLOCATE AUXILIARY CHANNEL ch3 DEVICE TYPE DISK;
  DUPLICATE TARGET DATABASE TO 'TEST'
  FROM ACTIVE DATABASE
  LOGFILE
    GROUP 1 ( '/u02/oradata/TEST/redog1m1.dbf' ,
              '/u03/oradata/TEST/redog1m2.dbf' ) SIZE 10M,
    GROUP 2 ( '/u02/oradata/TEST/redog2m1.dbf' ,
              '/u03/oradata/TEST/redog2m2.dbf' ) SIZE 10M;
}
```

# Duplicate without Target Connect (1)



- Backup based duplication techniques :
  - With Target Connect + with Catalog Connect
  - With Target Connect + without Catalog Connect
  - Without Target Connect + with Catalog Connect **New**
  - Without Target Connect + without Catalog Connect **New**
  
- Without Target and Catalog Connect the new Duplicate BACKUP LOCATION option must be specified :



```

DUPLICATE DATABASE to 'TECHDUP'
  BACKUP LOCATION '/u00/app/oracle/admin/TECH11/backup/'
  SPFILE
    PARAMETER_VALUE_CONVERT 'TECH11', 'TECHDUP'
    SET DB_FILE_NAME_CONVERT 'TECH11','TECHDUP'
    SET LOG_FILE_NAME_CONVERT 'TECH11','TECHDUP'
    SET CONTROL_FILES '/u01/oradata/TECHDUP/ct11TECHDUP.dbf ';
  
```

## Duplicate without Target Connect (2)



- Without Target and Catalog Connect the UNTIL clause can be used only with TIME

```
DUPLICATE DATABASE TO 'TECHDUP'
  BACKUP LOCATION '/u00/app/oracle/admin/TECH11/backup/'
  UNTIL TIME
    "to_date('29.10.09 02:06','dd.mm.yy hh24:mi')"
```

- Specify the target database name in the Duplicate command if:
  - There are backups of multiple databases in the BACKUP LOCATION
  - Without target but with Catalog Connect

```
DUPLICATE DATABASE TECH11 TO 'TECHDUP'
```

- Database in Catalog not unique:

```
DUPLICATE DATABASE TECH11 TO 'TECHDUP'
  DBID 2650824592 #DBID of the target database
```

## Duplicate without Target Connect (3)



- Example without Target and Catalog connect

```
CONNECT AUXILIARY sys/manager
DUPLICATE DATABASE TO 'TECHDUP'
  BACKUP LOCATION '/u00/app/oracle/admin/TECH11/backup/'
  SPFILE
  PARAMETER_VALUE_CONVERT 'TECH11', 'TECHDUP'
  SET DB_FILE_NAME_CONVERT 'TECH11', 'TECHDUP'
  SET LOG_FILE_NAME_CONVERT 'TECH11', 'TECHDUP'
  SET CONTROL_FILES
    '/u01/oradata/TECHDUP/ctl1TECHDUP.dbf',
    '/u02/oradata/TECHDUP/ctl2TECHDUP.dbf',
    '/u01/oradata/TECHDUP/ctl3TECHDUP.dbf';
```



# Data Recovery Advisor (1)



- Usable via Database Control and RMAN
- Failure – Status:
  - Open: detected Failure
  - Closed: Failure, with executed Repair Operation
- Failure – Priorities:
  - Critical: complete Database unavailable
  - High: Database partially unavailable and recoverable
  - Low: manual set through downgrade
- Display failures

```
LIST FAILURE;  
LIST FAILURE CLOSED;  
LIST FAILURE 7899 CLOSED DETAIL;
```

- Status and priority can be changed by CHANGE command

## Data Recovery Advisor (2)



- Manual or automatic Failure-Repair is possible
- Displays repair options including ADVISE commands

```
advise failure 7899;
```

- If repair is possible a suitable RMAN-script is created

### Automated Repair Options

=====

Option Repair Description

-----

1           Restore and recover datafile 4

    Strategy: The repair includes complete media recovery with no data loss

    Repair script:

```
/u00/app/oracle/diag/rdbms/pkndup1/PKNDUP1/hm/reco_2450428452.hm
```

## Data Recovery Advisor - Testresults(3)



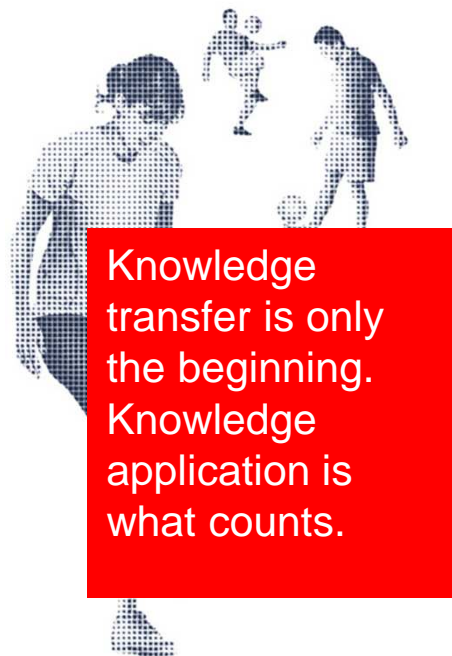
- Problems during spfile restore: Database is started via a dummy SPFILE and restored with a controlfile in ORACLE\_HOME/dbs
- Backups on different devices are ignored
- No failover on older backups are possible
- If automatic repair is not possible no hint is displayed

### Mandatory Manual Actions

=====

1. If file /u01/oradata/PKNDUP1/users01\_PKNDUP1.dbf was unintentionally renamed or moved, restore it
2. Contact Oracle Support Services if the preceding recommendations cannot be used, or if they do not fix the failures selected for repair

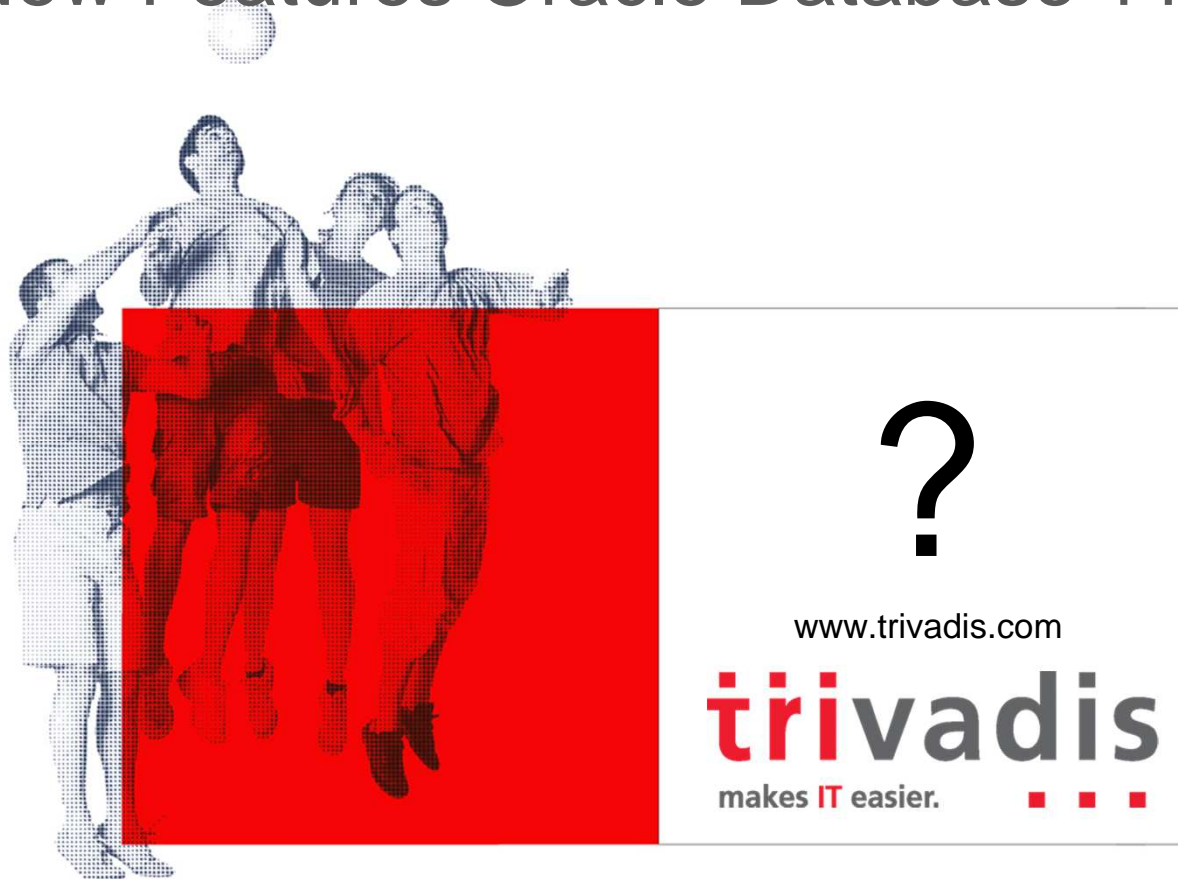
# Backup and Recovery – Core Messages



Knowledge transfer is only the beginning. Knowledge application is what counts.

- 😊 Multi section backups are in the area of bigfiles overdue
- 😊 Handling of RMAN scripts and catalog is more flexibel
- 😐 Backup compression with ZLIB is good, but combined with licence costs
- 😊 RMAN and DataGuard is more clear
- 😊 Active Database Duplication can be helpful in environments with many clone instances
- 😞 The idea of Data Recovery Advisor is super, but just at the beginning

# ■ ■ ■ Backup & Recovery New Features Oracle Database 11g



■ ■ ■ Thank You

