

```
SQL> @dgnfia
Connected.
----- Apply Lag Control
----- Make sure RTQ is active
SQL> @sysdba_kokki
SQL> connect sys/oracle@kokki as sysdba
Connected.
SQL> alter database open;
```

Database altered.

```
----- See the Data Guard config
SQL> !./dg_show_config
DGMGRL> show configuration;
```

Configuration - PeppiEnKokki

```
Protection Mode: MaxPerformance
Databases:
  peppi - Primary database
  kokki - Physical standby database
```

Fast-Start Failover: DISABLED

```
Configuration Status:
SUCCESS
```

```
----- Have a look at kokki
SQL> !./dg_show_database kokki
DGMGRL> show database kokki
```

Database - kokki

```
Role:                PHYSICAL STANDBY
Intended State:      APPLY-ON
Transport Lag:       0 seconds
Apply Lag:           0 seconds
Real Time Query:    ON
Instance(s):
  v1120
```

```
Database Status:
SUCCESS
```

```
----- See the apply lag in the database
SQL> select value, unit from v$dataguard_stats where name = 'apply lag';
```

```
VALUE          UNIT
-----
+00 00:00:00 day(2) to second(0) interval
```

```
----- Create an apply lag
SQL> !./dg_apply_off kokki
DGMGRL> edit database kokki set state=apply-off;
```

```
----- xmit some redo
```

```
SQL> @sysdba_peppi
SQL> connect sys/oracle@peppi as sysdba
Connected.
SQL> update scott.emp set sal = sal;
```

14 rows updated.

```
SQL> rollback;
```

Rollback complete.

----- Connect to kokki again

```
SQL> @sysdba_kokki
SQL> connect sys/oracle@kokki as sysdba
Connected.
```

----- See the apply lag

```
SQL> select value from v$dataguard_stats where name = 'apply lag';
```

VALUE

```
-----
+00 00:00:15
```

----- Set the standby_max_data_delay parameter

```
SQL> alter session set standby_max_data_delay = 60;
ERROR:
ORA-03174: STANDBY_MAX_DATA_DELAY does not apply to SYS users
```

----- OK, fair enough

```
SQL> connect system/oracle@kokki
Connected.
SQL> alter session set standby_max_data_delay = 60;
```

Session altered.

----- Perform a query

```
SQL> select count(*) from hr.employees;
```

```
   COUNT(*)
-----
         107
```

```
SQL> rem wait for a while
```

```
SQL> !sleep 30
```

----- Query again

```
SQL> select count(*) from hr.employees;
select count(*) from hr.employees
                        *
```

```
ERROR at line 1:
ORA-03172: STANDBY_MAX_DATA_DELAY of 60 seconds exceeded
```

----- See the apply lag again

```
SQL> select value from v$dataguard_stats where name = 'apply lag';
select value from v$dataguard_stats where name = 'apply lag'
*
```

```
ERROR at line 1:
```

ORA-00604: error occurred at recursive SQL level 1
ORA-03172: STANDBY_MAX_DATA_DELAY of 60 seconds exceeded

----- Hmmmm
----- Unset the standby_max_data_delay parameter
SQL> alter session set standby_max_data_delay = none;

Session altered.

SQL> select value from v\$dataguard_stats where name = 'apply lag';

VALUE

+00 00:01:29

----- Set the standby_max_data_delay parameter again
SQL> alter session set standby_max_data_delay = 60;

Session altered.

SQL> select count(*) from hr.employees;
select count(*) from hr.employees
*

ERROR at line 1:
ORA-03172: STANDBY_MAX_DATA_DELAY of 60 seconds exceeded

----- Force a sync with the primary
SQL> alter session sync with primary;
ERROR:
ORA-03173: Standby may not be synced with primary

----- Why
SQL> !./dg_apply_on kokki
DGMGRL> edit database kokki set state=apply-on;

----- Sync again
SQL> alter session sync with primary;

Session altered.

----- Query again
SQL> select count(*) from hr.employees;

COUNT(*)

107

----- If no delay is permitted, use
SQL> alter session set standby_max_data_delay = 0;

Session altered.

----- Query again
SQL> select count(*) from hr.employees;

COUNT(*)

107

----- Force an apply lag

```
SQL> !./dg_apply_off kokki
DGMGRL> edit database kokki set state=apply-off;
```

----- Query again

```
SQL> select count(*) from hr.employees;
select count(*) from hr.employees
*
ERROR at line 1:
ORA-03172: STANDBY_MAX_DATA_DELAY of 0 seconds exceeded
```

----- Enable redo apply again

```
SQL> !./dg_apply_on kokki
DGMGRL> edit database kokki set state=apply-on;
```

----- Press ENTER to continue.

```
SQL> select count(*) from hr.employees;
```

```
COUNT(*)
-----
107
```

----- How about a transport lag?

```
SQL> !./dg_ship_off kokki
DGMGRL> edit database kokki set property LogShipping=off;
Property "logshipping" updated
```

----- Query again

```
SQL> select count(*) from hr.employees;
select count(*) from hr.employees
*
ERROR at line 1:
ORA-03172: STANDBY_MAX_DATA_DELAY of 0 seconds exceeded
```

----- Enable redo shipping

```
SQL> !./dg_ship_on kokki
DGMGRL> edit database kokki set property LogShipping=on;
Property "logshipping" updated
```

----- Press ENTER to continue.

```
SQL> select count(*) from hr.employees;
```

```
COUNT(*)
-----
107
```

----- Clean up

```

----- Automatic Block Media Recovery
----- Create a tablespace to play with
SQL> connect system/oracle
Connected.
SQL> create tablespace corrupt datafile size 10m;

Tablespace created.

----- and a table
SQL> create table e1 tablespace corrupt as select * from scott.emp;

Table created.

----- Let us corrupt KING
SQL> select rowid from e1 where ename = 'KING';

ROWID
-----
AAADtaAAFAAAACDAAI

SQL> @CorruptRowBlock &rowid
SQL> --
SQL> -- Written by: Harald.van.Breederode@Oracle.com
SQL> --
SQL> set termout off
----- Make sure RTQ is off
SQL> !./dg_apply_off kokki
DGMGRL> edit database kokki set state=apply-off;

----- What does KING earn?
SQL> select sal from e1 where ename = 'KING';
select sal from e1 where ename = 'KING'
      *
ERROR at line 1:
ORA-01578: ORACLE data block corrupted (file # 5, block # 131)
ORA-01110: data file 5:
'/u01/db/v1120/data/PEPPI/datafile/o1_mf_corrupt_5j0915on_.dbf'

----- Put the physical standby into RTQ mode
SQL> !./dg_rtq_on kokki
DGMGRL> edit database kokki set state=apply-off;
edit database kokki set state= read-only;
edit database kokki set state=apply-on;

----- Create another table
SQL> create table e2 tablespace corrupt as select * from scott.emp;

Table created.

----- Let us corrupt JONES
SQL> select rowid from e2 where ename = 'JONES';

ROWID
-----

```

AAADtbAAFAAAACLAAD

```
SQL> @CorruptRowBlock &rowid
SQL> --
SQL> -- Written by: Harald.van.Breederode@Oracle.com
SQL> --
SQL> set termout off
----- What is JONES job?
SQL> select job from e2 where ename = 'JONES';
```

```
JOB
-----
MANAGER
```

```
----- Check the alert log
SQL> select value as trace from v$diag_info where name = 'Diag Trace';
```

```
TRACE
-----
-----
/u01/app/oracle/diag/rdbms/peppi/v1120/trace
```

```
SQL> !tail &trace/alert*.log
  spare1: 0x0 spare2: 0x0 spare3: 0x0
  consistency value in tail: 0x00000001
  check value in block header: 0xa78c
  computed block checksum: 0x0
Reading datafile
'/u01/db/v1120/data/PEPPI/datafile/ol_mf_corrupt_5j0915on_.dbf' for
corruption at rdba: 0x0140008b (file 5, block 139)
Reread (file 5, block 139) found same corrupt data
Requesting Auto BMR for (file# 5, block# 139)
Waiting Auto BMR response for (file# 5, block# 139)
Auto BMR successful
WARNING: AutoBMR fixed mismatched on-disk single block 8c with in-mem
rdba 140008b.
```

```
----- Clean up
SQL> drop table e1 purge;
```

Table dropped.

```
SQL> drop table e2 purge;
```

Table dropped.

```
SQL> drop tablespace corrupt;
```

Tablespace dropped.

----- Role Based Services

```
DGMGRL> edit database kokki set state=apply-on;
connect sys/oracle@kokki
startup force mount
ORACLE instance started.
Database mounted.
```

```
SQL> @sysdba_peppi
SQL> connect sys/oracle@peppi as sysdba
Connected.
```

----- See the Data Guard config

```
SQL> !./dg_show_config
DGMGRL> show configuration;
```

Configuration - PeppiEnKokki

```
Protection Mode: MaxPerformance
Databases:
  peppi - Primary database
  kokki - Physical standby database
```

Fast-Start Failover: DISABLED

```
Configuration Status:
SUCCESS
```

----- Register the primary database with Oracle Restart

```
SQL> !srvctl add db -d peppi -n v1120 -r primary -o $ORACLE_HOME
```

----- Display the Oracle Restart config for the primary

```
SQL> !srvctl config db -d peppi
Database unique name: peppi
Database name: v1120
Oracle home: /u01/app/oracle/product/11.2.0
Oracle user: oracle
Spfile:
Domain:
Start options: open
Stop options: immediate
Database role: primary
Management policy: AUTOMATIC
Disk Groups:
Services:
```

----- Let Oracle Restart discover the database status

```
SQL> !srvctl start db -d peppi
```

----- Register the physical standby with Oracle Restart

```
SQL> !ssh e15 ~/dg11gR2/run_with_env.sh srvctl add db -d kokki -n v1120
-r physical_standby -m mount -o $ORACLE_HOME
```

----- Display the Oracle Restart config for the physical standby

```
SQL> !ssh e15 ~/dg11gR2/run_with_env.sh srvctl config db -d kokki
Database unique name: kokki
```

Database name: v1120
Oracle home: /u01/app/oracle/product/11.2.0
Oracle user: oracle
Spfile:
Domain: mount
Start options: open
Stop options: immediate
Database role: physical_standby
Management policy: AUTOMATIC
Disk Groups:
Services:

----- Let Oracle Restart discover the database status
SQL> !ssh el5 ~/dg11gR2/run_with_env.sh srvctl start db -d kokki

----- And now, lets add the services
----- A service for production work
SQL> !srvctl add service -d peppi -s dg_prod -l primary -e select -m
basic -w 5 -z 50

----- See its config
SQL> !srvctl config service -d peppi -s dg_prod
Service name: dg_prod
Service is enabled
Cardinality: SINGLETON
Disconnect: false
Service role: primary
Management policy: AUTOMATIC
DTP transaction: false
AQ HA notifications: false
Failover type: SELECT
Failover method: BASIC
TAF failover retries: 50
TAF failover delay: 5
Connection Load Balancing Goal: LONG
Runtime Load Balancing Goal: NONE
TAF policy specification: NONE

----- And a service for RTQ
SQL> !srvctl add service -d peppi -s dg_rtq -l physical_standby -e
select -m basic -w 5 -z 50

----- See its config
SQL> !srvctl config service -d peppi -s dg_rtq
Service name: dg_rtq
Service is enabled
Cardinality: SINGLETON
Disconnect: false
Service role: physical_standby
Management policy: AUTOMATIC
DTP transaction: false
AQ HA notifications: false
Failover type: SELECT
Failover method: BASIC
TAF failover retries: 50
TAF failover delay: 5

Connection Load Balancing Goal: LONG
Runtime Load Balancing Goal: NONE
TAF policy specification: NONE

----- And one for RAT

```
SQL> !srvctl add service -d peppi -s dg_rat -l snapshot_standby
```

----- See its config

```
SQL> !srvctl config service -d peppi -s dg_rat
```

Service name: dg_rat
Service is enabled
Cardinality: SINGLETON
Disconnect: false
Service role: snapshot_standby
Management policy: AUTOMATIC
DTP transaction: false
AQ HA notifications: false
Failover type: NONE
Failover method: NONE
TAF failover retries: 0
TAF failover delay: 0
Connection Load Balancing Goal: LONG
Runtime Load Balancing Goal: NONE
TAF policy specification: NONE

----- Add them to EL5 as well!

----- The one for production work

```
SQL> !ssh e15 ~/dg11gR2/run_with_env.sh srvctl add service -d kokki -s  
dg_prod -l primary -e select -m basic -w 5 -z 50
```

----- See its config

```
SQL> !ssh e15 ~/dg11gR2/run_with_env.sh srvctl config service -d kokki -  
s dg_prod
```

Service name: dg_prod
Service is enabled
Cardinality: SINGLETON
Disconnect: false
Service role: primary
Management policy: AUTOMATIC
DTP transaction: false
AQ HA notifications: false
Failover type: SELECT
Failover method: BASIC
TAF failover retries: 50
TAF failover delay: 5
Connection Load Balancing Goal: LONG
Runtime Load Balancing Goal: NONE
TAF policy specification: NONE

----- The one for RTQ

```
SQL> !ssh e15 ~/dg11gR2/run_with_env.sh srvctl add service -d kokki -s  
dg_rtq -l physical_standby -e select -m basic -w 5 -z 50
```

----- See its config

```
SQL> !ssh e15 ~/dg11gR2/run_with_env.sh srvctl config service -d kokki -  
s dg_rtq
```

```
Service name: dg_rtq
Service is enabled
Cardinality: SINGLETON
Disconnect: false
Service role: physical_standby
Management policy: AUTOMATIC
DTP transaction: false
AQ HA notifications: false
Failover type: SELECT
Failover method: BASIC
TAF failover retries: 50
TAF failover delay: 5
Connection Load Balancing Goal: LONG
Runtime Load Balancing Goal: NONE
TAF policy specification: NONE
```

----- And the one for RAT

```
SQL> !ssh el5 ~/dg11gR2/run_with_env.sh srvctl add service -d kokki -s
dg_rat -l snapshot_standby
```

----- See its config

```
SQL> !ssh el5 ~/dg11gR2/run_with_env.sh srvctl config service -d kokki -s
dg_rat
```

```
Service name: dg_rat
Service is enabled
Cardinality: SINGLETON
Disconnect: false
Service role: snapshot_standby
Management policy: AUTOMATIC
DTP transaction: false
AQ HA notifications: false
Failover type: NONE
Failover method: NONE
TAF failover retries: 0
TAF failover delay: 0
Connection Load Balancing Goal: LONG
Runtime Load Balancing Goal: NONE
TAF policy specification: NONE
```

----- Is the DG_PROD service registered with the listener?

```
SQL> !lsnrctl services | grep -i dg_prod
```

----- Check the services in the dictionary

```
SQL> select name, failover_method, failover_type, failover_retries,
failover_delay
  2 from dba_services where name like 'dg_%';
```

no rows selected

----- Why are they not there?

```
SQL> !srvctl start service -d peppi -s dg_prod
```

----- Check the services in the dictionary again

```
SQL> select name, failover_method, failover_type, failover_retries,
failover_delay
  2 from dba_services where name like 'dg_%';
```

NAME	FAILOVER_METHOD	FAILOVER_TYPE	FAILOVER_RETRIES	FAILOVER_DELAY
dg_prod	BASIC	SELECT	50	5

----- Check the listener again

```
SQL> !lsnrctl services | grep -i dg_prod
Service "dg_prod" has 1 instance(s).
```

----- See the PROD TNS entry

```
SQL> !ft -f ^PROD -t ^# $TNS_ADMIN/tnsnames.ora
```

```
PROD = (DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = PRUTSER) (PORT = 1521))
  (ADDRESS = (PROTOCOL = TCP) (HOST = EL5) (PORT = 1521))
  (CONNECT_DATA = (SERVICE_NAME = DG_PROD)))
```

----- Connect using this entry

```
SQL> connect system/oracle@prod
Connected.
```

----- Where did we end up?

```
SQL> select host_name from v$instance;
```

```
HOST_NAME
```

```
-----
prutser.nl.oracle.com
```

----- See the TAF settings

```
SQL> select failover_method, failover_type, failed_over
  2 from v$session where username = 'SYSTEM';
```

FAILOVER_METHOD	FAILOVER_TYPE	FAI
BASIC	SELECT	NO

----- Perform a switchover

```
SQL> !./dg_switchover kokki
DGMGRL> switchover to kokki;
Performing switchover NOW, please wait...
New primary database "kokki" is opening...
Operation requires shutdown of instance "v1120" on database "peppi"
Shutting down instance "v1120"...
ORA-01109: database not open
```

Database dismounted.

ORACLE instance shut down.

Operation requires startup of instance "v1120" on database "peppi"

Starting instance "v1120"...

ORACLE instance started.

Database mounted.

Switchover succeeded, new primary is "kokki"

----- Are we still connected?

```
SQL> select failover_method, failover_type, failed_over
  2 from v$session where username = 'SYSTEM';
```

FAILOVER_METHOD	FAILOVER_TYPE	FAI
-----------------	---------------	-----

```
-----  
BASIC          SELECT          YES
```

```
----- On which host are we now connected?
```

```
SQL> select host_name from v$instance;
```

```
HOST_NAME
```

```
-----  
el5.nl.oracle.com
```

```
----- See the RAT TNS entry
```

```
SQL> !ft -f ^RAT -t ^# $TNS_ADMIN/tnsnames.ora
```

```
RAT = (DESCRIPTION =  
      (ADDRESS = (PROTOCOL = TCP) (HOST = PRUTSER) (PORT = 1521))  
      (ADDRESS = (PROTOCOL = TCP) (HOST = EL5) (PORT = 1521))  
      (CONNECT_DATA = (SERVICE_NAME = DG_RAT)))
```

```
----- Try an RAT connection
```

```
SQL> connect system/oracle@rat
```

```
ERROR:
```

```
ORA-12514: TNS:listener does not currently know of service requested in  
connect  
descriptor
```

```
Warning: You are no longer connected to ORACLE.
```

```
----- OK, lets fix
```

```
SQL> !./dg_snapshot_on @kokki peppi
```

```
DGMGRL> convert database peppi to snapshot standby;
```

```
Converting database "peppi" to a Snapshot Standby database, please  
wait...
```

```
Database "peppi" converted successfully
```

```
----- Show the Data Guard config
```

```
SQL> !./dg_show_config
```

```
DGMGRL> show configuration;
```

```
Configuration - PeppiEnKokki
```

```
Protection Mode: MaxPerformance
```

```
Databases:
```

```
kokki - Primary database
```

```
peppi - Snapshot standby database
```

```
Fast-Start Failover: DISABLED
```

```
Configuration Status:
```

```
SUCCESS
```

```
----- Is the service running?
```

```
SQL> !srvctl status service -d peppi -s dg_rat
```

```
Service dg_rat is running
```

```
----- Check the listener
```

```
SQL> !lsnrctl services | grep -i dg_rat
```

Service "dg_rat" has 1 instance(s).

----- retry to connect

SQL> connect system/oracle@rat

Connected.

----- Did we end up where we should?

SQL> select database_role from v\$database;

DATABASE_ROLE

SNAPSHOT STANDBY

----- Check the services in the dictionary

SQL> select name, failover_method, failover_type, failover_retries,
failover_delay
2 from dba_services where name like 'dg_%';

NAME	FAILOVER_METHOD	FAILOVER_TYPE	FAILOVER_RETRIES	FAILOVER_DELAY
dg_prod	BASIC	SELECT	50	5
dg_rat	NONE	NONE	0	0

----- Think about this!

----- See the RTQ TNS entry

SQL> !ft -f ^RTQ -t ^# \$TNS_ADMIN/tnsnames.ora

```
RTQ = (DESCRIPTION =  
      (ADDRESS = (PROTOCOL = TCP) (HOST = PRUTSER) (PORT = 1521))  
      (ADDRESS = (PROTOCOL = TCP) (HOST = EL5) (PORT = 1521))  
      (CONNECT_DATA = (SERVICE_NAME = DG_RTQ)))
```

----- Try a RTQ connection

SQL> connect system/oracle@rtq

ERROR:

ORA-12514: TNS:listener does not currently know of service requested in
connect
descriptor

Warning: You are no longer connected to ORACLE.

----- ok, make one

----- First get rid of the snapshot standby

SQL> !./dg_snapshot_off @kokki peppi

DGMGRL> convert database peppi to physical standby;

Converting database "peppi" to a Physical Standby database, please
wait...

Operation requires shutdown of instance "v1120" on database "peppi"

Shutting down instance "v1120"...

Database closed.

Database dismounted.

ORACLE instance shut down.

Operation requires startup of instance "v1120" on database "peppi"

Starting instance "v1120"...

ORACLE instance started.

Database mounted.

Continuing to convert database "peppi" ...

Operation requires shutdown of instance "v1120" on database "peppi"

Shutting down instance "v1120"...
ORA-01109: database not open

Database dismounted.
ORACLE instance shut down.
Operation requires startup of instance "v1120" on database "peppi"
Starting instance "v1120"...
ORACLE instance started.
Database mounted.
Database "peppi" converted successfully

----- Second make sure RTQ is on
SQL> @sysdba_peppi
SQL> connect sys/oracle@peppi as sysdba
Connected.
SQL> alter database open;

Database altered.

----- Show the Data Guard config
SQL> !./dg_show_config
DGMGRL> show configuration;

Configuration - PeppiEnKokki

Protection Mode: MaxPerformance
Databases:
kokki - Primary database
peppi - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

----- Show the RTQ database
SQL> !./dg_show_database peppi
DGMGRL> show database peppi

Database - peppi

Role: PHYSICAL STANDBY
Intended State: APPLY-ON
Transport Lag: 0 seconds
Apply Lag: 0 seconds
Real Time Query: ON
Instance(s):
v1120

Database Status:
SUCCESS

----- Retry an RTQ connection
SQL> connect system/oracle@rtq

ERROR:
ORA-12514: TNS:listener does not currently know of service requested in connect descriptor

Warning: You are no longer connected to ORACLE.

----- Why?

----- Check the listener

SQL> !lsnrctl services | grep -i dg_rtq

----- Check the service

SQL> !srvctl status service -d peppi -s dg_rtq

Service dg_rtq is not running.

----- Check the services in the dictionary

SQL> @sysdba_peppi

SQL> connect sys/oracle@peppi as sysdba

Connected.

SQL> select name, failover_method, failover_type, failover_retries, failover_delay

2 from dba_services where name like 'dg_%';

NAME	FAILOVER_METHOD	FAILOVER_TYPE	FAILOVER_RETRIES	FAILOVER_DELAY
dg_prod	BASIC	SELECT	50	5

----- Try to start the RTQ service manually

SQL> !srvctl start service -d peppi -s dg_rtq

PRCR-1079 : Failed to start resource ora.peppi.dg_rtq.svc

ORA-44317: database open read-only

ORA-06512: at "SYS.DBMS_SERVICE", line 469

ORA-06512: at "SYS.DBMS_SERVICE", line 222

ORA-06512: at line 1

CRS-2674: Start of 'ora.peppi.dg_rtq.svc' on 'prutser' failed

ORA-44304: service dg_rtq does not exist

ORA-06512: at "SYS.DBMS_SYS_ERROR", line 86

ORA-06512: at "SYS.DBMS_SERVICE", line 443

ORA-06512: at "SYS.DBMS_SERVICE", line 391

ORA-06512: at line 1

----- How can we add the RTQ service in the data dictionary?

SQL> !ssh e15 ~/dg11gR2/run_with_env.sh srvctl start service -d kokki -s dg_rtq

SQL> !ssh e15 ~/dg11gR2/run_with_env.sh srvctl stop service -d kokki -s dg_rtq

----- Check the services in the dictionary

SQL> @sysdba_peppi

SQL> connect sys/oracle@peppi as sysdba

Connected.

SQL> select name, failover_method, failover_type, failover_retries, failover_delay

```
2 from dba_services where name like 'dg_%';
```

NAME	FAILOVER_METHOD	FAILOVER_TYPE	FAILOVER_RETRIES	FAILOVER_DELAY
dg_prod	BASIC	SELECT	50	5
dg_rtq	BASIC	SELECT	50	5

```
----- Start the RTQ service manually
SQL> !srvctl start service -d peppi -s dg_rtq
```

```
----- Check the listener
SQL> !lsnrctl services | grep -i dg_rtq
Service "dg_rtq" has 1 instance(s).
```

```
----- Retry an RTQ connection
SQL> connect system/oracle@rtq
Connected.
```

```
----- Where did we end up?
SQL> select host_name from v$instance;
```

```
HOST_NAME
```

```
-----
prutser.nl.oracle.com
```

```
----- Perform a switchover
SQL> !./dg_switchover peppi
DGMGRL> switchover to peppi;
Performing switchover NOW, please wait...
New primary database "peppi" is opening...
Operation requires shutdown of instance "v1120" on database "kokki"
Shutting down instance "v1120"...
ORA-01109: database not open
```

```
Database dismounted.
ORACLE instance shut down.
Operation requires startup of instance "v1120" on database "kokki"
Starting instance "v1120"...
ORACLE instance started.
Database mounted.
Database opened.
Switchover succeeded, new primary is "peppi"
```

```
----- Are we still connected?
SQL> select host_name from v$instance;
```

```
HOST_NAME
```

```
-----
el5.nl.oracle.com
```

```
----- Can we still connect to do production work?
SQL> connect system/oracle@prod
Connected.
```

```
----- Where did we end up?
SQL> select host_name from v$instance;
```

```
HOST_NAME
```

prutser.nl.oracle.com

----- Clean up

SQL> !srvctl stop service -d peppi -s dg_prod

SQL> !srvctl remove db -d peppi -f -v

Successfully removed database and its dependent services.

SQL> !ssh e15 ~/dg11gR2/run_with_env.sh srvctl stop service -d kokki -s
dg_rtq

SQL> !ssh e15 ~/dg11gR2/run_with_env.sh srvctl remove db -d kokki -f -v
Successfully removed database and its dependent services.

SQL> @sysdba_peppi

SQL> connect sys/oracle@peppi as sysdba

Connected.

SQL> exec dbms_service.delete_service('dg_prod')

PL/SQL procedure successfully completed.

SQL> exec dbms_service.delete_service('dg_rtq')

PL/SQL procedure successfully completed.