

ORACLE®

Introduction to Data Guard

NY SIG Meeting

October 7th, 2003

Who am I?

Mr. Paranoid

(It's my job)

Larry M. Carpenter

Senior Principal Consultant

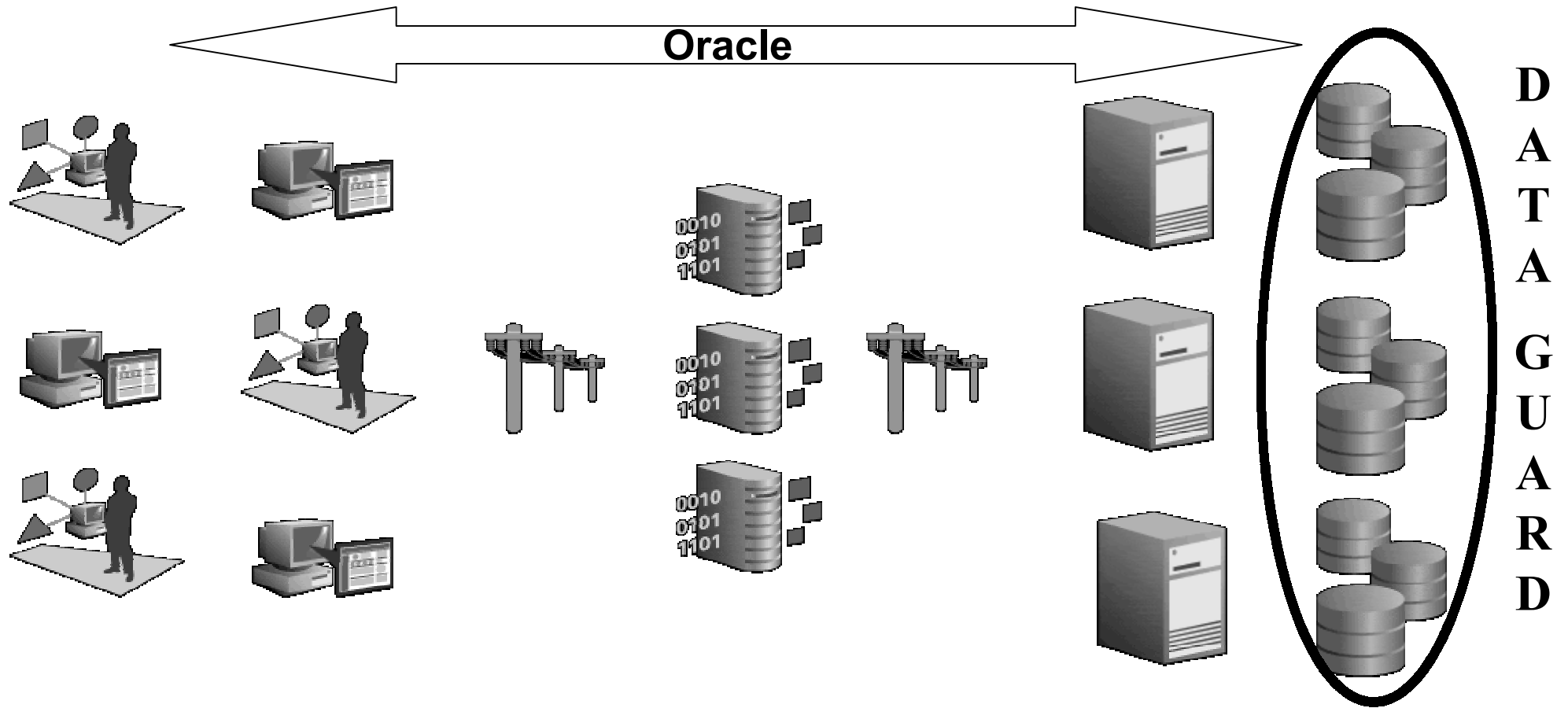
Data Guard Development

Server Technologies

Oracle Corporation

What is Data Guard?

Disaster Recovery Food Chain



Users

Networks

Applications

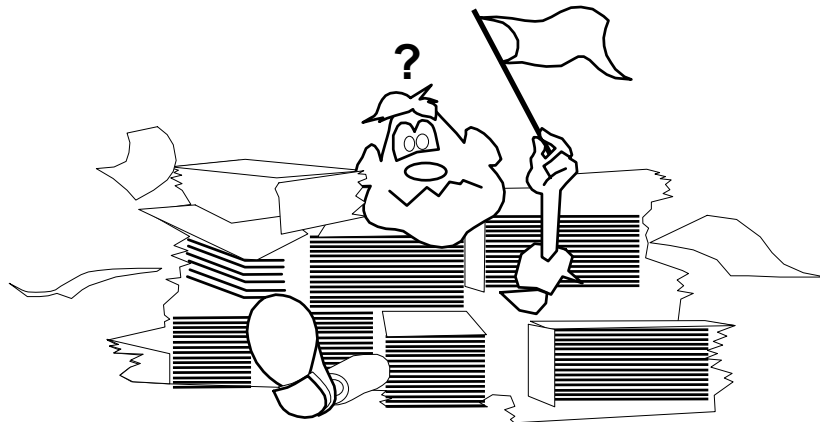
Servers

Databases

ORACLE

So, just what is Data Guard?

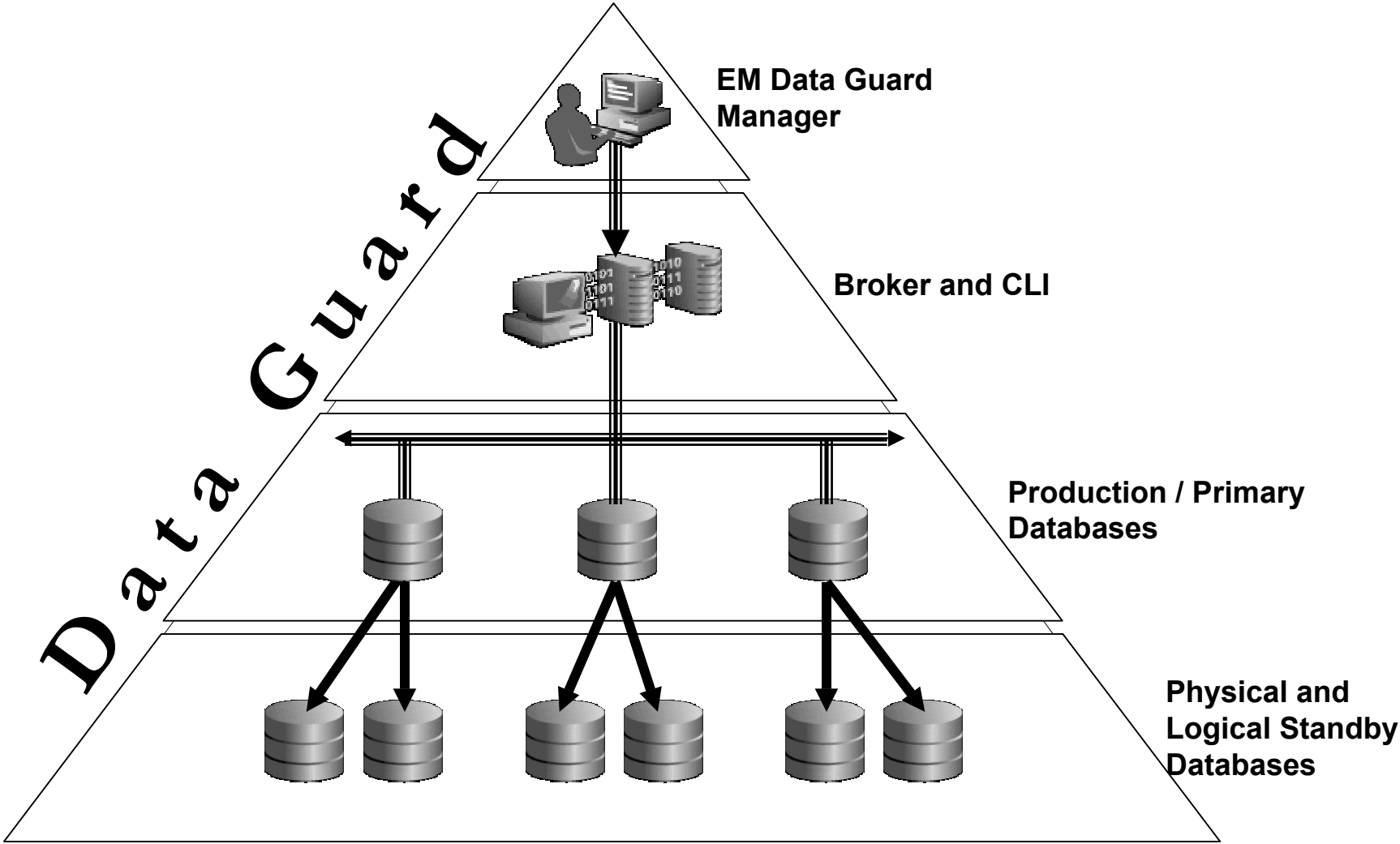
- “An application-transparent high-performance low-impact asymmetrical online reliable Redo or SQL level background standby database transaction exchange utility capable of reporting, switchover and Failover.”
- What?



Simply put...

- Data Guard helps you protect your Data.
 - Takes your data and automatically puts it elsewhere
 - Makes it available for Failover in case of failure.
- The other capabilities are pure bonus.
 - Switchover for Maintenance
 - Reporting
 - Off-loading Queries
 - Backups

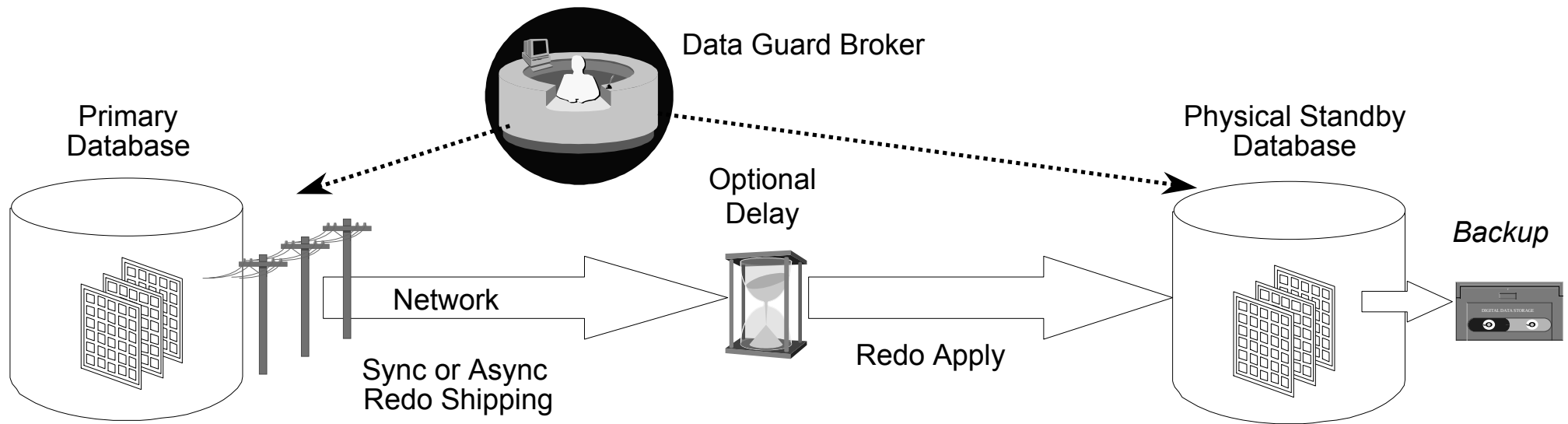
Data Guard 'Pyramid'



High Level

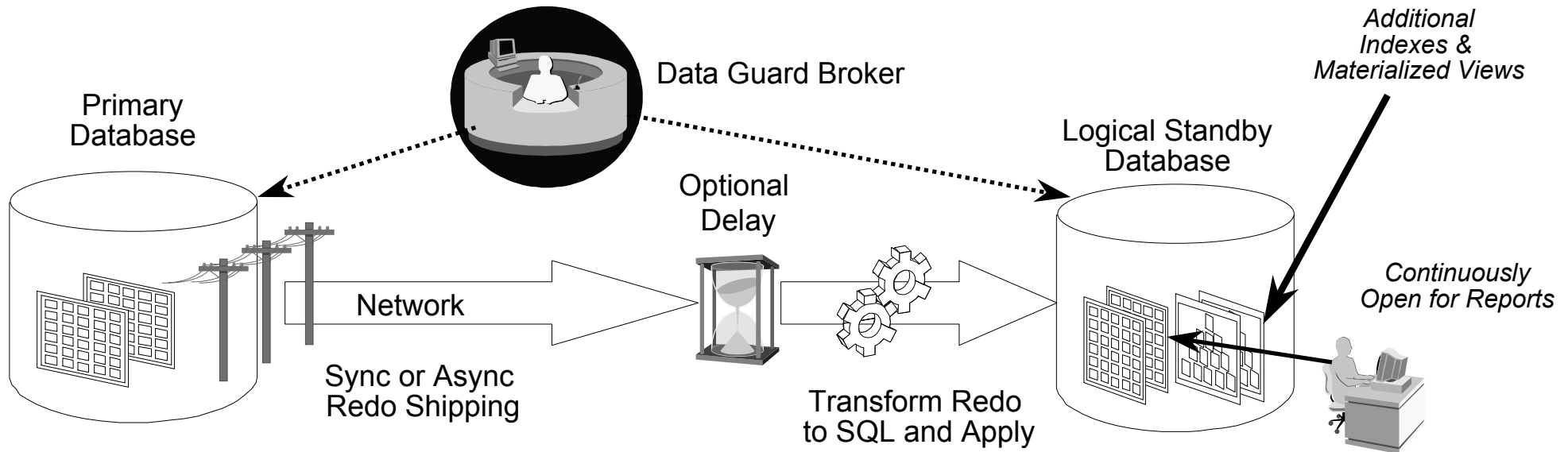
- Data Guard comprises of two parts
 - **REDO APPLY (DR)**
 - Maintains a physical, block for block copy of the Production (also called Primary) database.
 - **SQL APPLY (Reporting)**
 - Maintains a logical, transaction for transaction copy of the Production database.

Data Guard Redo Apply: *Best for DR*



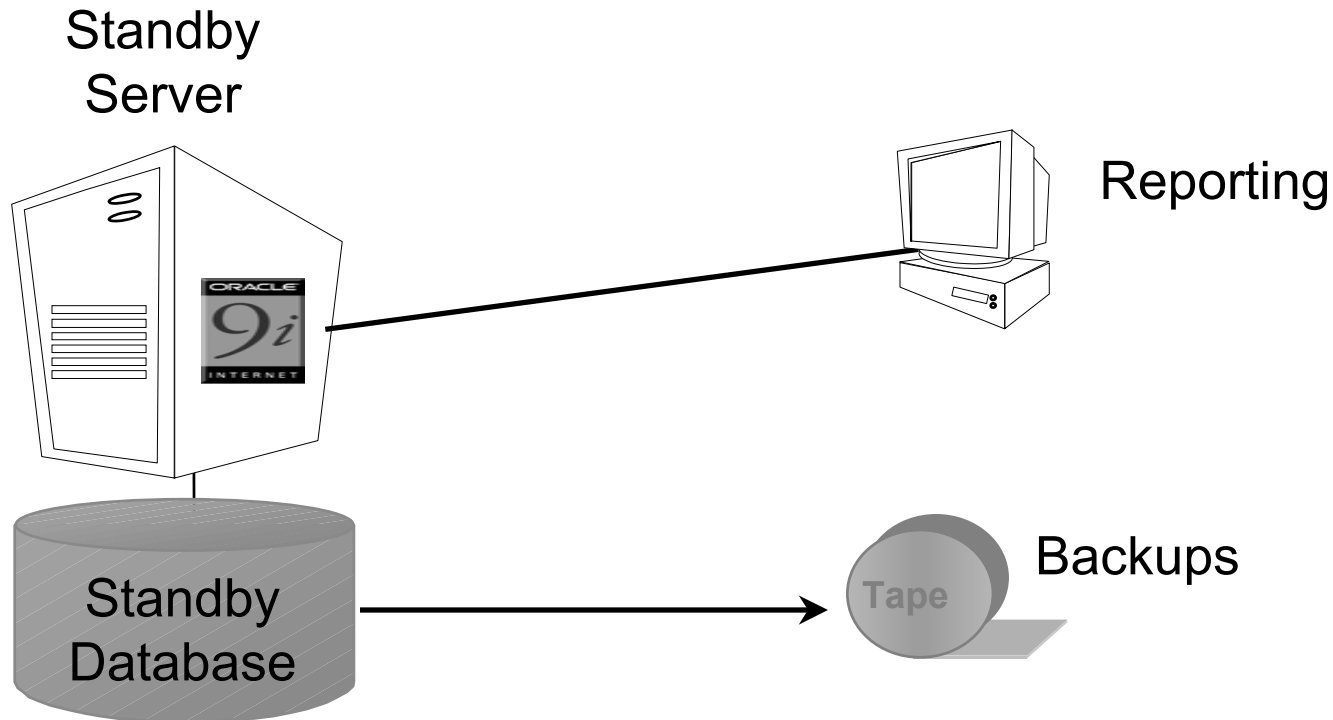
- Physical Standby Database is a block-for-block copy of the primary database
- Uses the database recovery functionality to apply changes
- Can be opened in read-only mode for reporting/queries
- Can also perform backup, offloading production database
- The best solution for DR

Data Guard SQL Apply



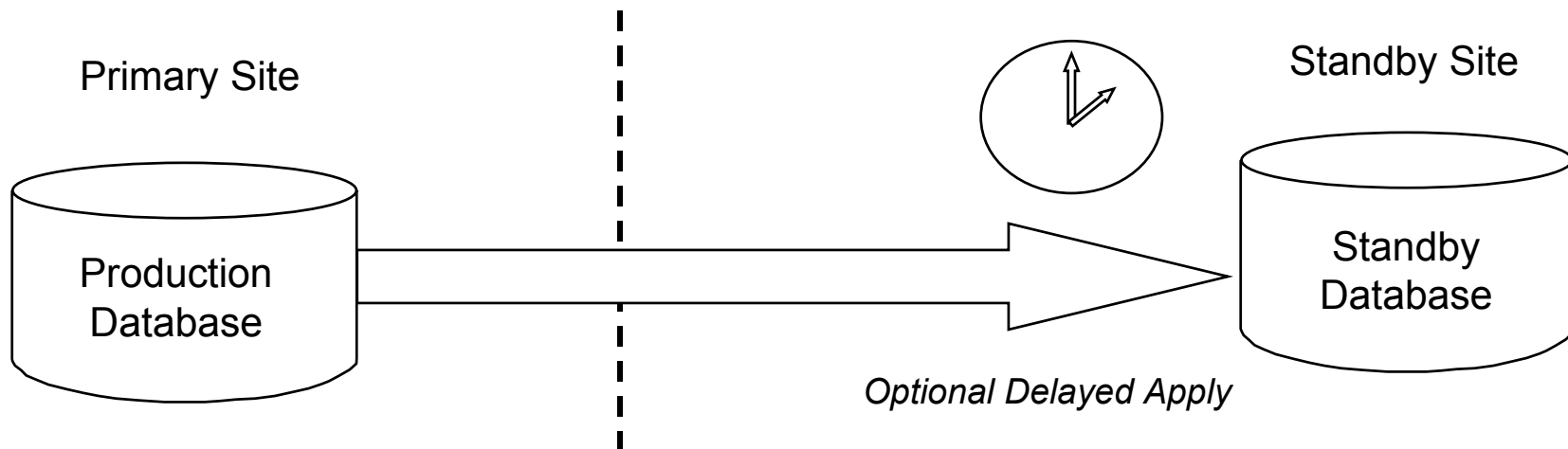
- Logical Standby Database is an open, independent, active database
 - Contains the same logical information (rows) as the production database
 - Physical organization and structure can be very different
 - Can host multiple schemas
- Can be queried for reports while logs are being applied via SQL
- Can create additional indexes and materialized views for better query performance
- Not all Data Types supported (See the manual for a list)

Standby Databases Are Not Idle



Standby database can be used to offload the primary database, increasing the ROI

Protection from Human Errors and Data Corruptions



- The application of changes received from the primary can be delayed at standby to allow for the detection of user errors and prevent standby to be affected
- The apply process also revalidates the log records to prevent application of any log corruptions

Before We Get Started

TANSTAAFL

There Ain't No Such Thing As A Free Lunch!

'The Moon is a Harsh Mistress' – Robert Heinlein

So, How Easy is it to Setup?

Not Rocket Science!

- "Data Guard now has many sophisticated DR/HA features, but still the thing that impresses me the most is its ease of implementation and long term reliability. We don't have to baby sit it. If there are problems, we don't have to dig through documentation to remember how it works. Our management has told us to do more with less DBAs, and Data Guard has helped us implement a solid DR/HA solution without adding DBAs."
 - Darl Kuhn – Lead DBA Sun IT

Setup Overview

- Step 1 - Prepare the Primary for Standby
- Step 2 - Copy the necessary files to standby system
- Step 3 - Configure the Standby Parameters
- Step 4 - Configure OracleNet
- Step 5 - Startup the Standby Site
- Step 6 - Begin Shipping and Applying Redo

Setup the Production Database

```
Terminal
Window Edit Options Help
SQL*Plus: Release 9.2.0.1.0 - Production on Mon Feb 3 16:12:55 2003
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.
I
Connected to:
Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production
SQL> select name,database_role from v$database;
NAME          DATABASE_ROLE
-----
PAYROLL       PRIMARY
SQL> archive log list;
Database log mode                Archive Mode
Automatic archival                Enabled
Archive destination              /private2/oracle/OraHome92010/oradata/Payroll/Archive
Oldest online log sequence       2
Next log sequence to archive     4
Current log sequence              4
SQL> █
```

Check Archiving and Force Logging

```
Terminal
Window Edit Options Help
SQL>
SQL>
SQL>
SQL>
SQL> show parameter log_archive_dest_1
NAME                                TYPE                                VALUE
-----                                -                                -
log_archive_dest_1                  string                              LOCATION=/private2/oracle/OraH
ome92010/oradata/Payroll/Archi
ve
log_archive_dest_10                 string
SQL> show parameter log_archive_dest_2
NAME                                TYPE                                VALUE
-----                                -                                -
log_archive_dest_2                  string
SQL>
SQL>
SQL> alter database force logging;
Database altered.
SQL>
```

Copy the Data files to the Standby

```
Terminal
Window Edit Options Help
SQL> select name from v$datafile;
NAME
-----
/private2/oracle/OraHome92010/oradata/Payroll/system01.dbf
/private2/oracle/OraHome92010/oradata/Payroll/undotbs01.dbf
/private2/oracle/OraHome92010/oradata/Payroll/example01.dbf
/private2/oracle/OraHome92010/oradata/Payroll/indx01.dbf
/private2/oracle/OraHome92010/oradata/Payroll/tools01.dbf
/private2/oracle/OraHome92010/oradata/Payroll/users01.dbf

6 rows selected.

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> ! rcp $ORACLE_HOME/oradata/Payroll/*.dbf 1carpent2:$ORACLE_HOME/oradata/Payroll/

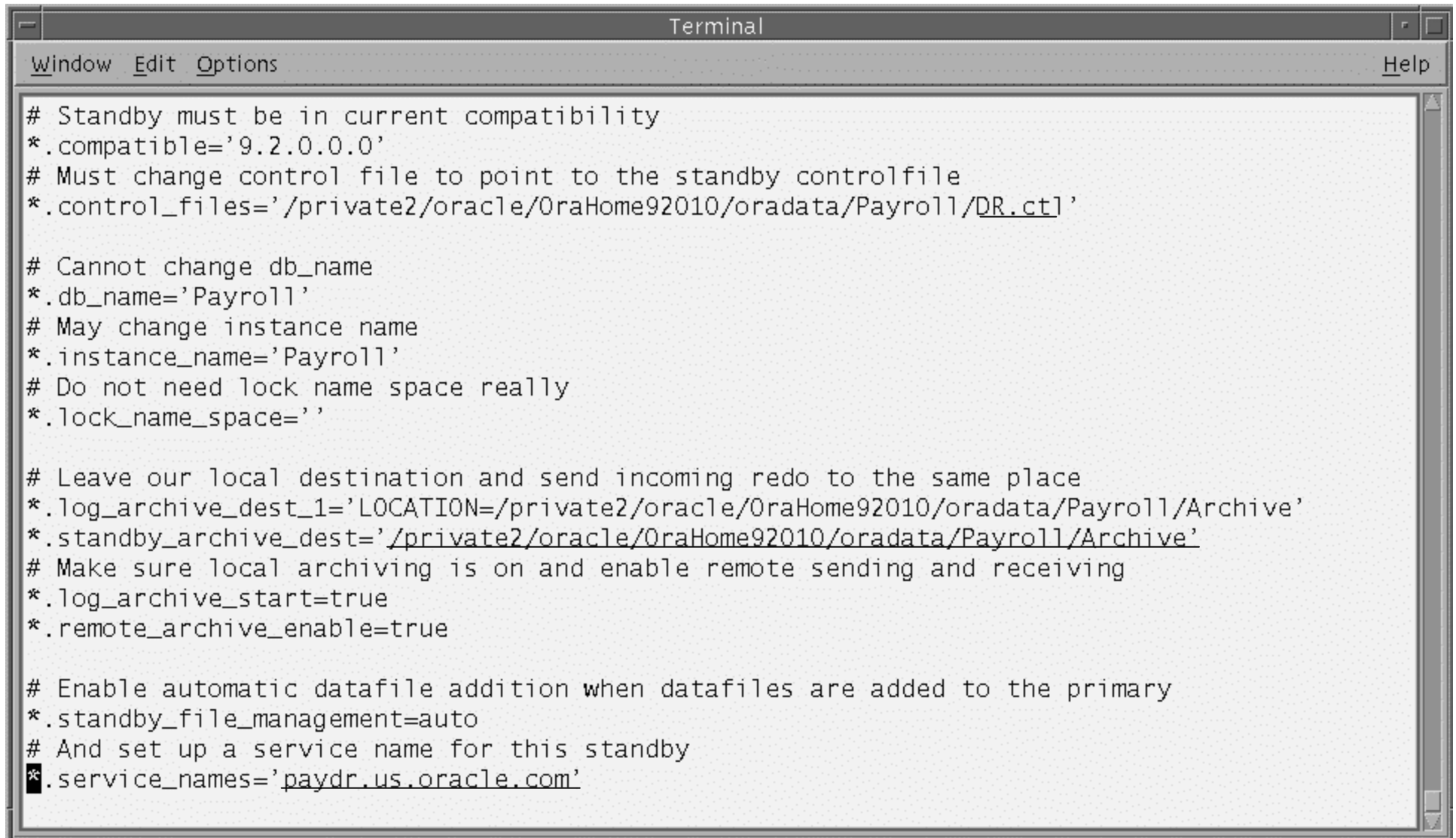
SQL> startup
ORACLE instance started.

Total System Global Area  110186876 bytes
Fixed Size                  455036 bytes
```

Standby Control file and Init file

```
Terminal
Window Edit Options Help
SQL> alter database create standby controlfile as '$ORACLE_HOME/oradata/Payroll/DR.ct1';
Database altered.
SQL> create pfile='$ORACLE_HOME/oradata/Payroll/initPayroll.ora' from spfile;
File created.
SQL> ! rcp $ORACLE_HOME/oradata/Payroll/initPayroll.ora 1carpent2:$ORACLE_HOME/dbs/
SQL> ! rcp $ORACLE_HOME/oradata/Payroll/DR.ct1 1carpent2:$ORACLE_HOME/oradata/Payroll/
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
```

Setup the Standby Init Parameters

A terminal window titled "Terminal" with a menu bar containing "Window", "Edit", "Options", and "Help". The terminal displays a list of Oracle standby initialization parameters. Each parameter is preceded by a comment line starting with "#". The parameters are: compatible, control_files, db_name, instance_name, lock_name_space, log_archive_dest_1, standby_archive_dest, log_archive_start, remote_archive_enable, standby_file_management, and service_names.

```
Terminal
Window Edit Options Help
# Standby must be in current compatibility
*.compatible='9.2.0.0.0'
# Must change control file to point to the standby controlfile
*.control_files='/private2/oracle/OraHome92010/oradata/Payroll/DR.ctl'

# Cannot change db_name
*.db_name='Payroll'
# May change instance name
*.instance_name='Payroll'
# Do not need lock name space really
*.lock_name_space=''

# Leave our local destination and send incoming redo to the same place
*.log_archive_dest_1='LOCATION=/private2/oracle/OraHome92010/oradata/Payroll/Archive'
*.standby_archive_dest='/private2/oracle/OraHome92010/oradata/Payroll/Archive'
# Make sure local archiving is on and enable remote sending and receiving
*.log_archive_start=true
*.remote_archive_enable=true

# Enable automatic datafile addition when datafiles are added to the primary
*.standby_file_management=auto
# And set up a service name for this standby
*.service_names='paydr.us.oracle.com'
```


Setup the Production Side TNS

```
Terminal
Window Edit Options Help

PAYROLLDR =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 1carpent2.us.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = paydr.us.oracle.com)
    )
  )

PAYROLL =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 1carpent-sun)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = 1carpent2)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = Payroll.us.oracle.com)
    )
  )

1carpent-sun> █
```

Setup the Standby Side TNS

```
Terminal
Window Edit Options Help
# Generated by Oracle configuration tools.

PAYROLLDR =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 1carpent-sun.us.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = paydr.us.oracle.com)
    )
  )

PAYROLL =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 1carpent2.us.oracle.com)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = 1carpent-sun.us.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = payroll.us.oracle.com)
    )
  )

1carpent2> █
```

Launch the Standby Database

```
Terminal
Window Edit Options Help
lcarpent2> orapwd file=$ORACLE_HOME/dbs/orapwPayroll password=oracle
lcarpent2> sqlplus "sys/oracle as sysdba"

SQL*Plus: Release 9.2.0.1.0 - Production on Tue Feb 4 16:49:22 2003
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to an idle instance.

SQL> startup nomount
ORACLE instance started.

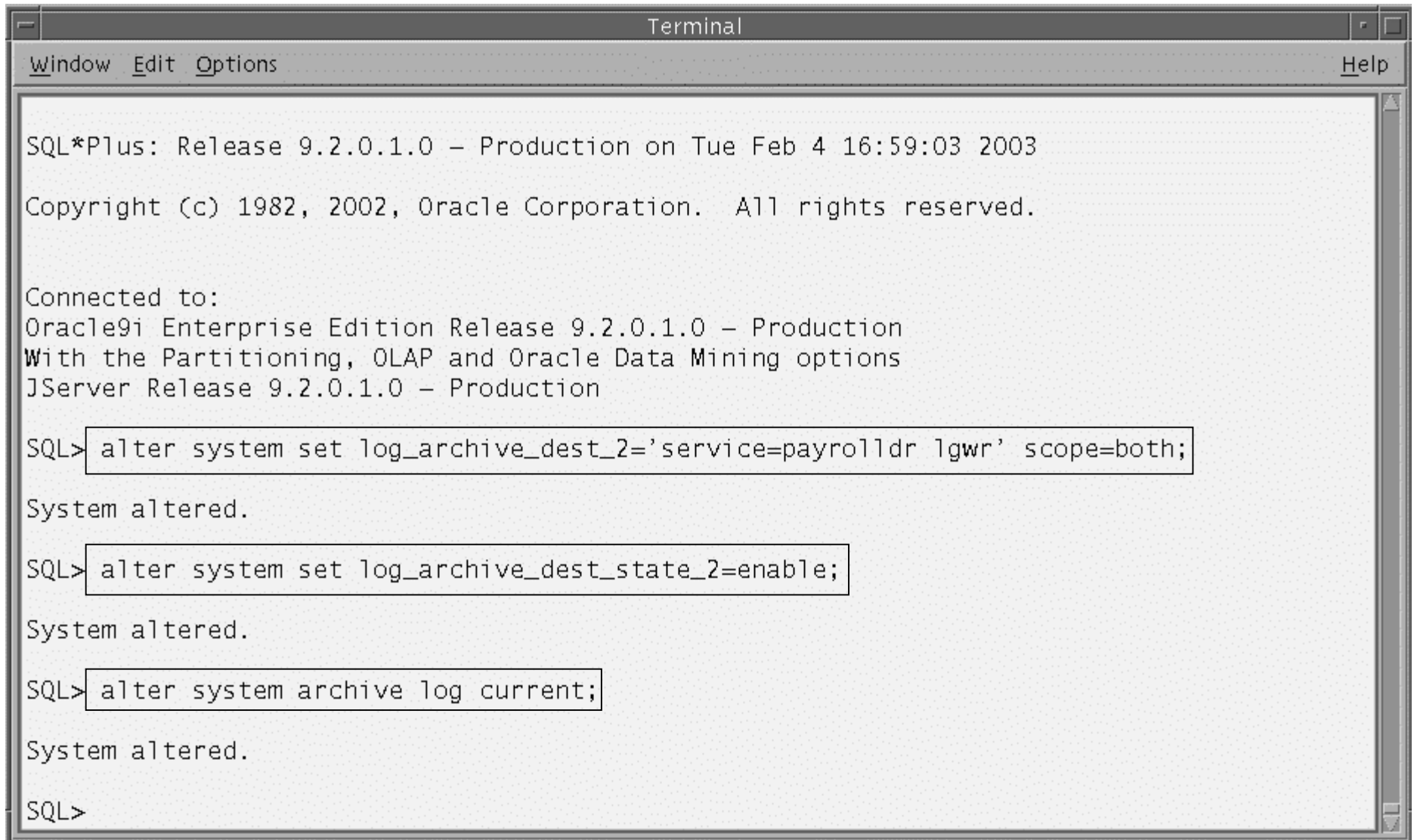
Total System Global Area  110186876 bytes
Fixed Size                  455036 bytes
Variable Size              92274688 bytes
Database Buffers           16777216 bytes
Redo Buffers                 679936 bytes
SQL> alter database mount standby database;

Database altered.

SQL> alter database recover managed standby database disconnect from session;

Database altered.
```

Start Sending Redo!



```
Terminal
Window Edit Options Help

SQL*Plus: Release 9.2.0.1.0 - Production on Tue Feb 4 16:59:03 2003
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production

SQL> alter system set log_archive_dest_2='service=payrolldr lgwr' scope=both;
System altered.

SQL> alter system set log_archive_dest_state_2=enable;
System altered.

SQL> alter system archive log current;
System altered.

SQL>
```

Verify the Primary is sending Redo

```
Terminal
Window Edit Options Help
SQL> select dest_id,sequence#,archived,applied from v$archived_log order by 1,2;

```

DEST_ID	SEQUENCE#	ARC	APP
1	2	YES	NO
1	3	YES	NO
1	4	YES	NO
2	4	YES	YES

```
SQL> alter system archive log current;
System altered.
SQL> select dest_id,sequence#,archived,applied from v$archived_log order by 1,2;

```

DEST_ID	SEQUENCE#	ARC	APP
1	2	YES	NO
1	3	YES	NO
1	4	YES	NO
1	5	YES	NO
2	4	YES	YES
2	5	YES	NO

Add in the Standby Redo Log Files

```
Terminal
Window Edit Options Help
SQL> alter database recover managed standby database cancel;
Database altered.
SQL> select substr(f.member,1,60) "File",l.bytes "Size" from v$logfile f, v$log l where f.group#=l.group#;
File
-----
/private2/oracle/OraHome92010/oradata/Payroll/redo01.log      10485760
/private2/oracle/OraHome92010/oradata/Payroll/redo02.log      10485760
/private2/oracle/OraHome92010/oradata/Payroll/redo03.log      10485760
SQL> alter database add standby logfile '$ORACLE_HOME/oradata/Payroll/sr101.log' size 10m;
Database altered.
SQL> alter database add standby logfile '$ORACLE_HOME/oradata/Payroll/sr102.log' size 10m;
Database altered.
SQL> alter database add standby logfile '$ORACLE_HOME/oradata/Payroll/sr103.log' size 10m;
Database altered.
```

Make sure they are being used

```
SQL> alter system archive log current;
```

System altered.

```
SQL> select group#,sequence#,status from v$log;
```

GROUP#	SEQUENCE#	STATUS
1	5	INACTIVE
2	6	ACTIVE
3	7	CURRENT

On the Primary

```
SQL> █
```

```
SQL> alter database recover managed standby database disconnect from session;
```

Database altered.

```
SQL> select group#,sequence#,status from v$standby_log;
```

GROUP#	SEQUENCE#	STATUS
4	7	ACTIVE
5	0	UNASSIGNED
6	0	UNASSIGNED

On the Standby

```
SQL> █
```

We're Done!

- Well, I thought that was easy.

Changing Roles

Switchover and Failover

- There are two ways to change roles in a standby configuration
 - Switchover
 - Changing roles with someone else and letting them take over while you become a standby
 - Failover
 - Assigning someone else to take over when the original boss is gone
- Different steps for Physical and Logical Standby
- We'll do a Physical Standby Switchover

Prepare the Primary Parameters

```
Terminal
Window Edit Options Help

SQL> show parameter standby

NAME                                TYPE      VALUE
-----                                -
standby_archive_dest                 string    ?/dbs/arch
standby_file_management               string    MANUAL
SQL> alter system set standby_archive_dest='&ORACLE_HOME/oradata/Payroll/Archive' scope=spfile;

System altered.

SQL> alter system set standby_file_management=AUTO scope=spfile;

System altered.

SQL> show parameter remote_archive

NAME                                TYPE      VALUE
-----                                -
remote_archive_enable                string    true
SQL>
SQL>
SQL> █
```

Prepare the Standby Parameters

```
Terminal
Window Edit Options Help
SQL>
SQL>
SQL>
SQL>
SQL> show parameter log_archive_dest_2
NAME                                TYPE                                VALUE
-----                                -                                -
log_archive_dest_2                   string
SQL>
SQL> alter system set log_archive_dest_2='service=payrolldr lgwr' scope=spfile;
System altered.
SQL> alter system set log_archive_dest_state_2=DEFER scope=spfile;
System altered.
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
```

Prepare to Switchover the Primary

```
Terminal
Window Edit Options Help
SQL>
SQL>
SQL> select switchover_status from v$database;
SWITCHOVER_STATUS
-----
SESSIONS ACTIVE
SQL> select username,status,process,type from v$session where type='USER';
USERNAME          STATUS  PROCESS  TYPE
-----
SYS                ACTIVE   4336     USER
SYS                INACTIVE 4922     USER
SQL> select switchover_status from v$database;
SWITCHOVER_STATUS
-----
TO STANDBY
SQL>
SQL>
SQL> █
```

Start with the Primary

```
Terminal
Window Edit Options Help

SWITCHOVER_STATUS
-----
TO STANDBY

SQL>
SQL>
SQL> alter database commit to switchover to physical standby;

Database altered.

SQL> select switchover_status from v$database;
select switchover_status from v$database
          *
ERROR at line 1:
ORA-01507: database not mounted

SQL> shutdown immediate
ORA-01507: database not mounted

ORACLE instance shut down.
SQL> █
```

Don't do this until the standby has received all the redo!

Then Switchover the Standby

```
Terminal
Window Edit Options Help

SQL> select switchover_status from v$database;

SWITCHOVER_STATUS
-----
NOT ALLOWED

SQL> select switchover_status from v$database;

SWITCHOVER_STATUS
-----
TO PRIMARY

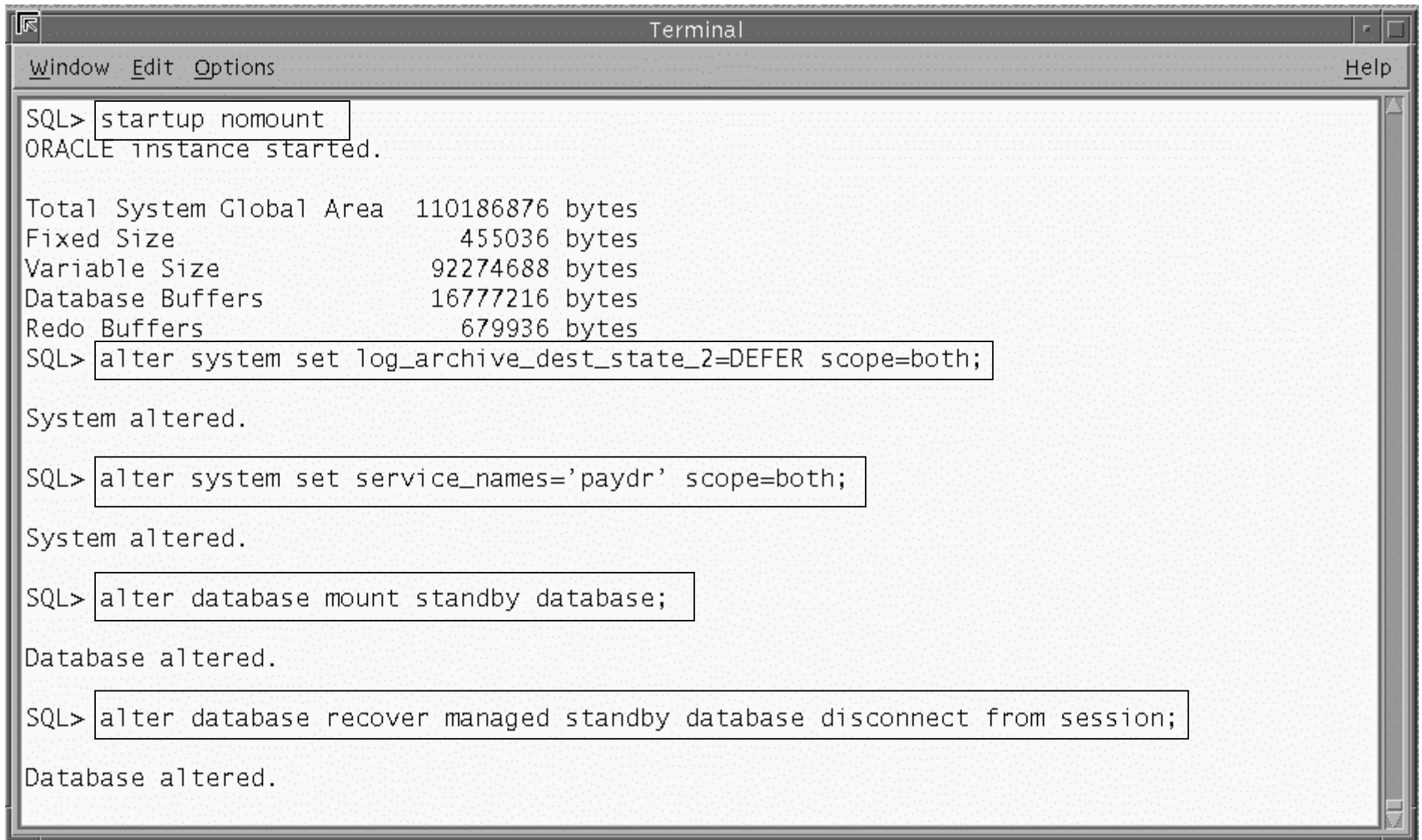
SQL> alter database commit to switchover to primary;

Database altered.

SQL> shutdown immediate
ORA-01507: database not mounted

ORACLE instance shut down.
SQL> █
```

Startup the New Standby

A terminal window titled "Terminal" with a menu bar containing "Window", "Edit", "Options", and "Help". The terminal displays the following SQL commands and their outputs:

```
SQL> startup nomount
ORACLE instance started.

Total System Global Area  110186876 bytes
Fixed Size                 455036 bytes
Variable Size             92274688 bytes
Database Buffers         16777216 bytes
Redo Buffers              679936 bytes
SQL> alter system set log_archive_dest_state_2=DEFER scope=both;

System altered.

SQL> alter system set service_names='paydr' scope=both;

System altered.

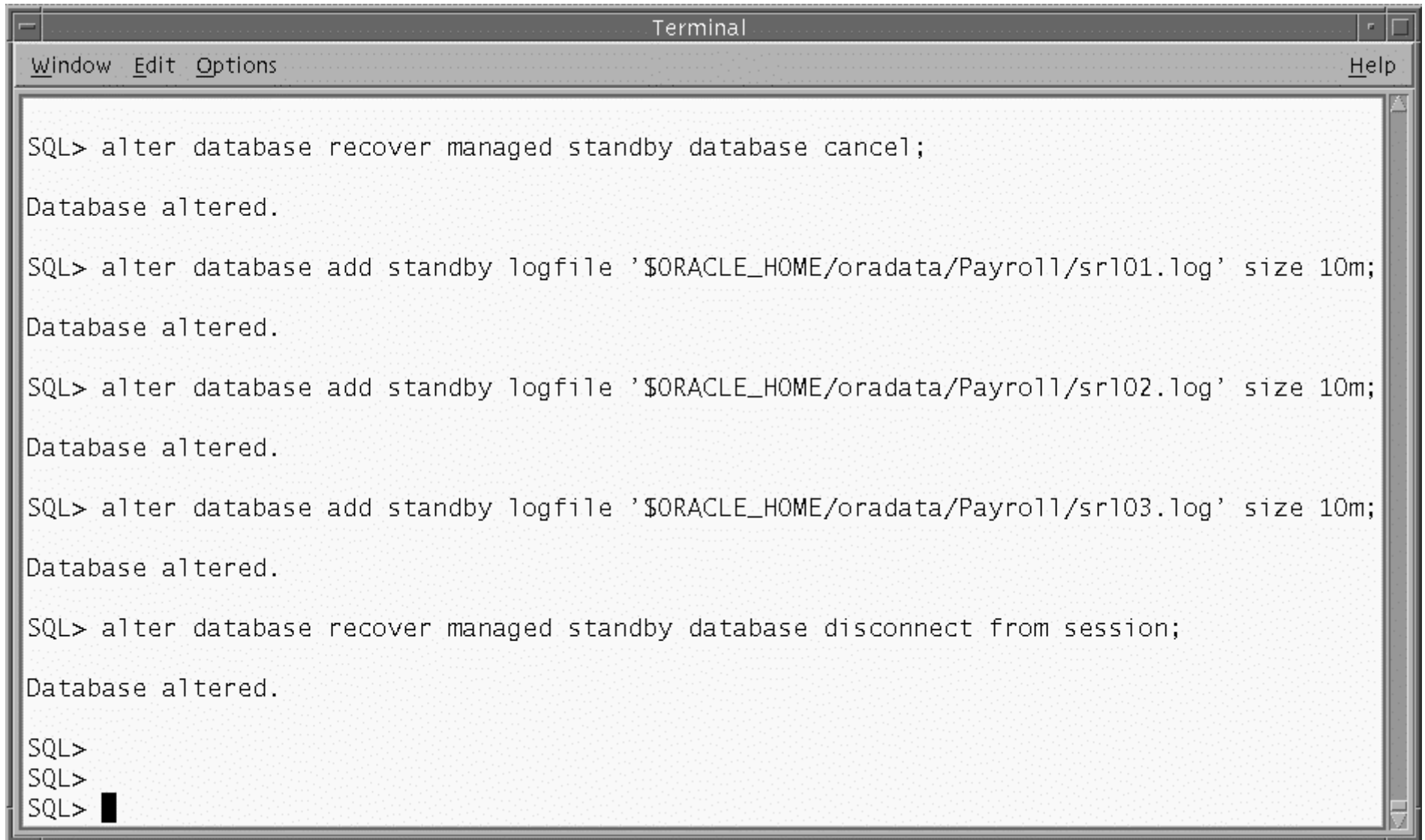
SQL> alter database mount standby database;

Database altered.

SQL> alter database recover managed standby database disconnect from session;

Database altered.
```


Add in the SRL's to the New Standby



```
Terminal
Window Edit Options Help

SQL> alter database recover managed standby database cancel;
Database altered.

SQL> alter database add standby logfile '$ORACLE_HOME/oradata/Payroll/sr101.log' size 10m;
Database altered.

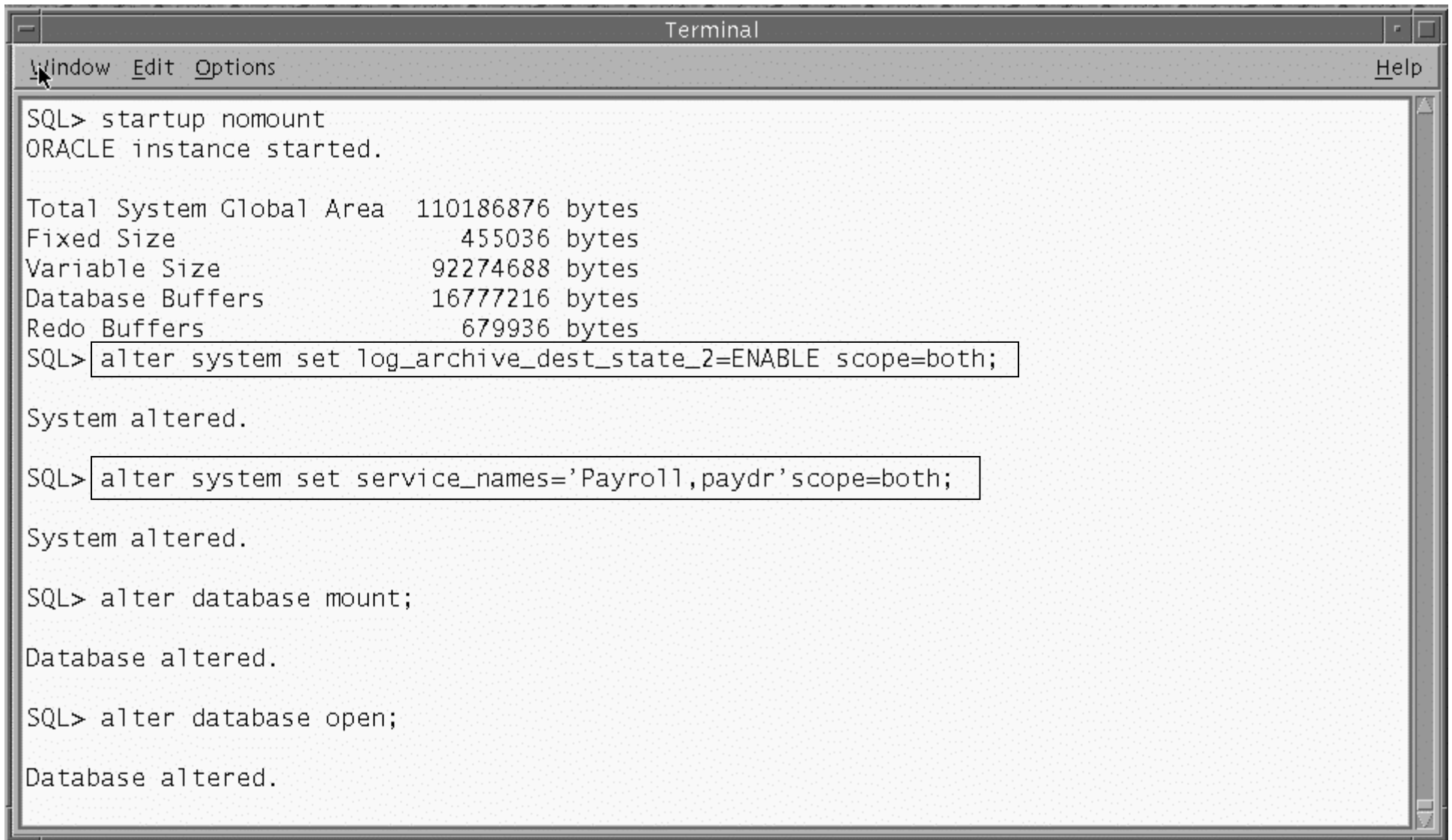
SQL> alter database add standby logfile '$ORACLE_HOME/oradata/Payroll/sr102.log' size 10m;
Database altered.

SQL> alter database add standby logfile '$ORACLE_HOME/oradata/Payroll/sr103.log' size 10m;
Database altered.

SQL> alter database recover managed standby database disconnect from session;
Database altered.

SQL>
SQL>
SQL> █
```

Startup the New Primary



```
Terminal
Window Edit Options Help
SQL> startup nomount
ORACLE instance started.

Total System Global Area  110186876 bytes
Fixed Size                 455036 bytes
Variable Size             92274688 bytes
Database Buffers         16777216 bytes
Redo Buffers               679936 bytes
SQL> alter system set log_archive_dest_state_2=ENABLE scope=both;

System altered.

SQL> alter system set service_names='Payroll,paydr' scope=both;

System altered.

SQL> alter database mount;

Database altered.

SQL> alter database open;

Database altered.
```

Verify the New Standby

```
Terminal
Window Edit Options Help

SQL>
SQL> select process,status,sequence# from v$managed_standby;

PROCESS STATUS          SEQUENCE#
-----
ARCH     CONNECTED           0
ARCH     CONNECTED           I 0
MRPO     WAIT_FOR_LOG        13
RFS      WRITING              13
RFS      RECEIVING           10
RFS      ATTACHED            11

6 rows selected.

SQL> select sequence#,status from v$standby_log;

SEQUENCE# STATUS
-----
13 ACTIVE
0 UNASSIGNED
0 UNASSIGNED

SQL> █
```

Verify the New Primary

```
Terminal
Window Edit Options Help

SQL> select SEQUENCE#,STATUS from v$log;

SEQUENCE# STATUS
-----
11 INACTIVE
13 CURRENT
12 INACTIVE

SQL> select dest_id,status from v$archive_dest_status where dest_id=2;

DEST_ID STATUS
-----
2 VALID

SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
```

“Switchback?”

- Just do the previous slides again!
 - Without all the parameters changes other than setting the service names and enabling or deferring the remote destinations.

Ok, now let's do a Failover!

- This will recover all of our data since I have it setup as a zero data loss configuration.
- The current Primary will have to be recreated after a Failover.

Insert Data and Crash the Primary

```
Terminal
Window Edit Options Help

1 create table testdata
2 (id number,
3* name varchar(20))
SQL> /

Table created.

SQL> insert into testdata values (1,'Larry Carpenter');

1 row created.
SQL> commit;
Commit complete.

SQL> shutdown abort
ORACLE instance shut down.
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
```

**No Log
Switch!**

Verify the Standby and Fail Over

```
Terminal
Window Edit Options Help
SQL> select * from v$archive_gap;
no rows selected
SQL> select process,client_process,sequence# from v$managed_standby;
PROCESS CLIENT_P SEQUENCE#
-----
ARCH     ARCH          0
ARCH     ARCH          0
MRPO     N/A           13
SQL> select switchover_status from v$database;
SWITCHOVER_STATUS
-----
NOT ALLOWED
SQL> alter database recover managed standby database finish;
Database altered.
SQL> █
```



Switch over to Primary

```
Terminal
Window Edit Options Help
SQL> select switchover_status from v$database;
SWITCHOVER_STATUS
-----
TO PRIMARY
SQL> alter database commit to switchover to primary;
Database altered.
SQL> shutdown immediate
ORA-01507: database not mounted
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

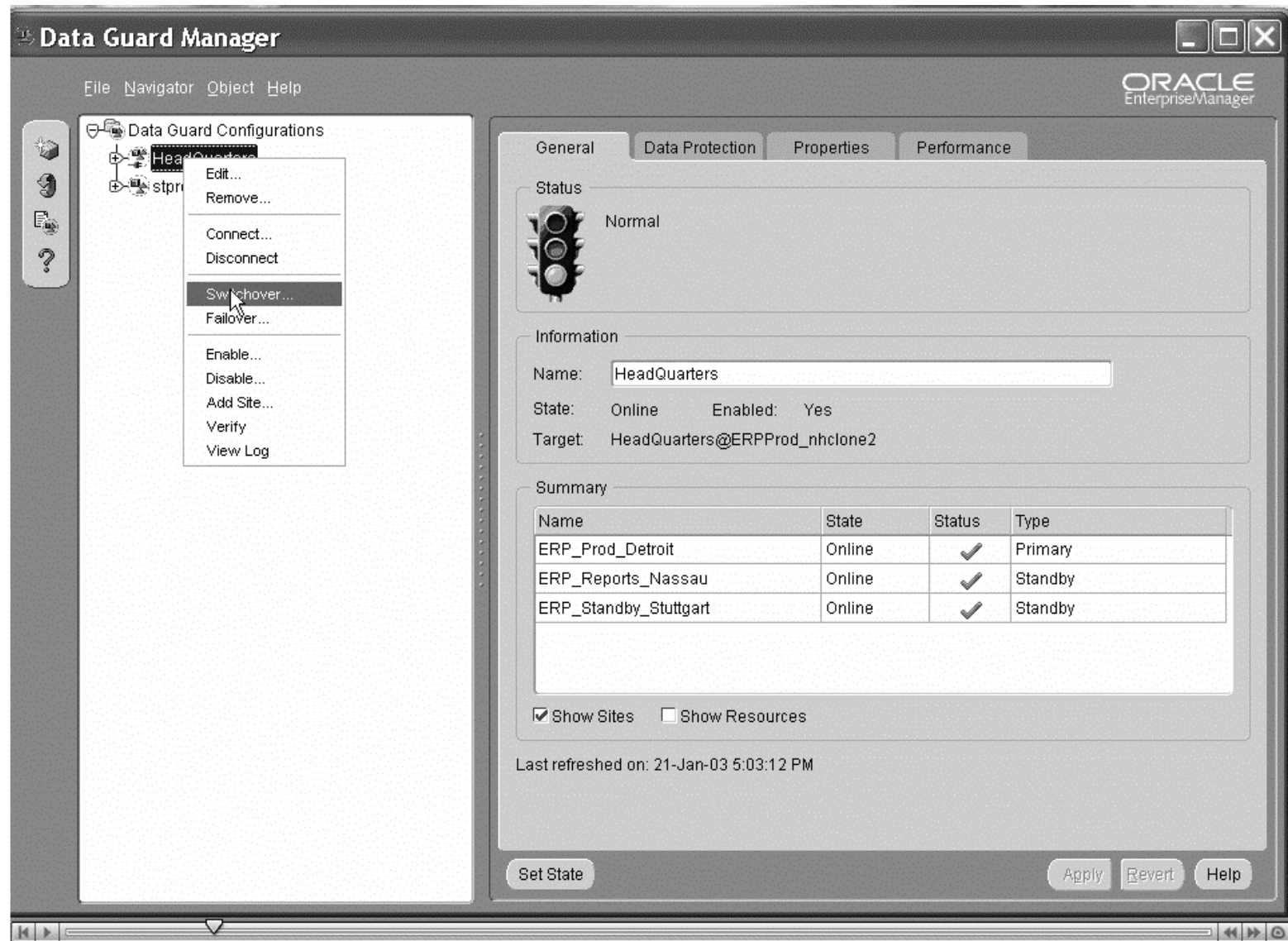
Total System Global Area 110186876 bytes
Fixed Size                 455036 bytes
Variable Size             92274688 bytes
Database Buffers         16777216 bytes
Redo Buffers              679936 bytes
Database mounted.
```

Setup Access and Verify Data

```
Terminal
Window Edit Options Help
SQL> show parameter service
NAME                                TYPE      VALUE
-----                                -
mts_service                          string    Payroll
service_names                        string    paydr
SQL>
SQL> alter system set service_names='Payroll,paydr' scope=both;
System altered.
SQL> desc testdata
Name                                Null?    Type
-----                                -
ID                                   NUMBER
NAME                                VARCHAR2(20)
SQL> select * from testdata;
   ID NAME
-----
    1 Larry Carpenter
.....
SQL> █
```



Of Course You Could use the GUI



Protection Levels

- Transport Services define how the redo gets to the standby site.
 - In Oracle 9i Release 1 that is all you had.
- The Protection Levels define how the Primary functions in the standby configuration
 - Maximize Protection
 - Maximize Availability
 - Maximize Performance
- Each one has a defined set of rules

Protection Modes

Protection Mode	Failure Protection	Redo Shipping
Maximum Protection Zero Data Loss	Protects Against Primary and Network Failure	LGWR using SYNC and SRL
Maximum Availability Zero Data Loss	Protects Against Primary Failure	LGWR using SYNC
Maximum Performance	Best Effort Against Primary Failure	ARCH or LGWR using ASYNC

Maximum Protection Mode

Protection Mode	Failure Protection	Redo Shipping
Maximum Protection Zero Data Loss	Protects Against Primary and Network Failure	LGWR using SYNC and SRL

- Zero Data Loss!
- Highest Level of Protection
- Configuration: LGWR SYNC, SRLs
- Enforces protection of every transaction
- If last standby is unavailable, processing stops at primary
- Good for financial systems where no data loss is acceptable

```
ALTER DATABASE SET STANDBY TO MAXIMIZE PROTECTION;
```

Maximum Availability Mode

Protection Mode	Failure Protection	Redo Shipping
Maximum Availability Zero Data Loss	Protects Against Primary Failure	LGWR using SYNC

- Zero Data Loss as long as the network stays up!
- Enforces protection of every transaction
- Configuration: LGWR SYNC, do not need SRLs
- If last standby is unavailable, processing continues at primary
- When the standby becomes available again, synchronization with the primary is automatic

```
ALTER DATABASE SET STANDBY TO MAXIMIZE AVAILABILITY;
```

Maximum Performance Mode

Protection Mode	Failure Protection	Redo Shipping
Maximum Performance	Best Effort Against Primary Failure	ARCH or LGWR using ASYNC

- Highest level of performance
- Configuration: LGWR ASYNC, or ARCH
- Protects from failure of any single component
- Least impact on production system
- Useful for applications that can tolerate some data loss

```
ALTER DATABASE SET STANDBY TO MAXIMIZE PERFORMANCE;
```


Data Guard and Oracle Apps 11i

- Data Guard standbys require redo in the log
 - No logging operations on the primary means missing data on the standbys.
 - Physical Standbys will work but any no logging operations by the Apps means exposure and manual operations to resynchronize
 - More information
 - MetaLink Note 216212.1 & 216211.1
 - Oracle 9.2 has Force Logging which will solve these issues
- Logical Standby will not work correctly
 - Missing critical data type support

Requirements

Installation and Configuration Considerations

- Enterprise Edition only for the Server
- Requires the same version and release of the Oracle database server for the primary and all standby sites.
 - Each primary database and standby database must have its own control file.
 - The primary database must run in ARCHIVELOG mode.
- Requires the same hardware architecture on the primary and all standby sites.
- Does not require the same version and release of the operating system on the primary and all standby sites.

Minimum Database Requirements

- What do you need at a minimum?
 - An Oracle9i primary database.
 - Release 1 – 9.0.1.3 or higher
 - Release 2 – 9.2.0.2 or higher if possible
 - There are several patches to 9.2.0.2 if you do not have 9.2.0.3
 - Trust me, you need them
 - At Oracle9i Release 2 if you want SQL Apply
 - A standby database
 - Same version as the primary
 - With *Standby Redo Logs* if it's a Physical standby

Minimum Environment Requirements

- What else do you need?
 - A network between the two!
 - Primary system tnsname to the standby listener
 - Standby system tnsname to the primary listener
 - If the pipe isn't big enough to send the redo it isn't going to work!
 - And no, I do not recommend sneaker net!
 - Redo Transport Services on the Primary
 - Defines how the redo gets shipped to the standby
 - A set of rules for the configuration to follow
 - Which defines how you expect it to operate

Some other Gotcha's

- Force Logging
 - If you are at Release 2 use the force logging command
 - ALTER DATABASE FORCE LOGGING;
 - If it isn't in the redo stream, it isn't in the standby.
- Know your Production Database!
 - If you are using a Physical standby everything is supported provided you force logging!
 - If you want to use a Logical standby there are several unsupported data types and other considerations

What About RAC and Streams?

Data Guard and RAC

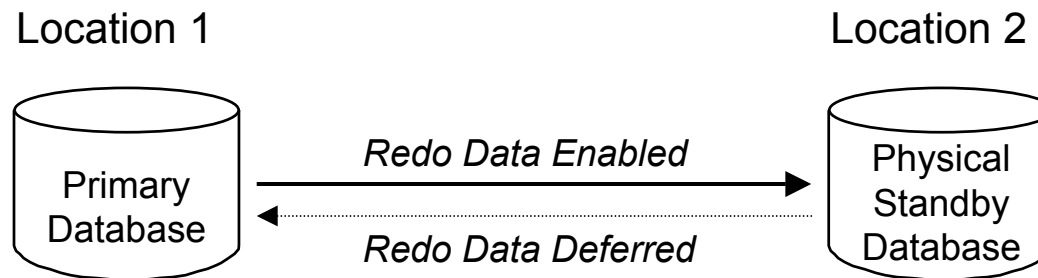
- **RAC**: high availability and scalability solution within a data center, implemented on a single set of storage
- **Data Guard**: Disaster Recovery and Data Protection solution that can span data centers, implemented on multiple storage systems
- Data Guard and RAC are complementary and should be used together as foundations of a Maximum Availability Architecture

Data Guard and Streams

- Streams and Data Guard are independent features of Oracle Database Enterprise Edition, based on some common underlying technology
- Data Guard: Disaster Recovery & Data Protection
 - Transactionally consistent standby databases
 - Zero data loss
- Streams: Information Sharing/Distribution
 - Fine granularity and control over what is replicated
 - Heterogeneous platforms

Some Configurations

Basic Physical Standby Configuration

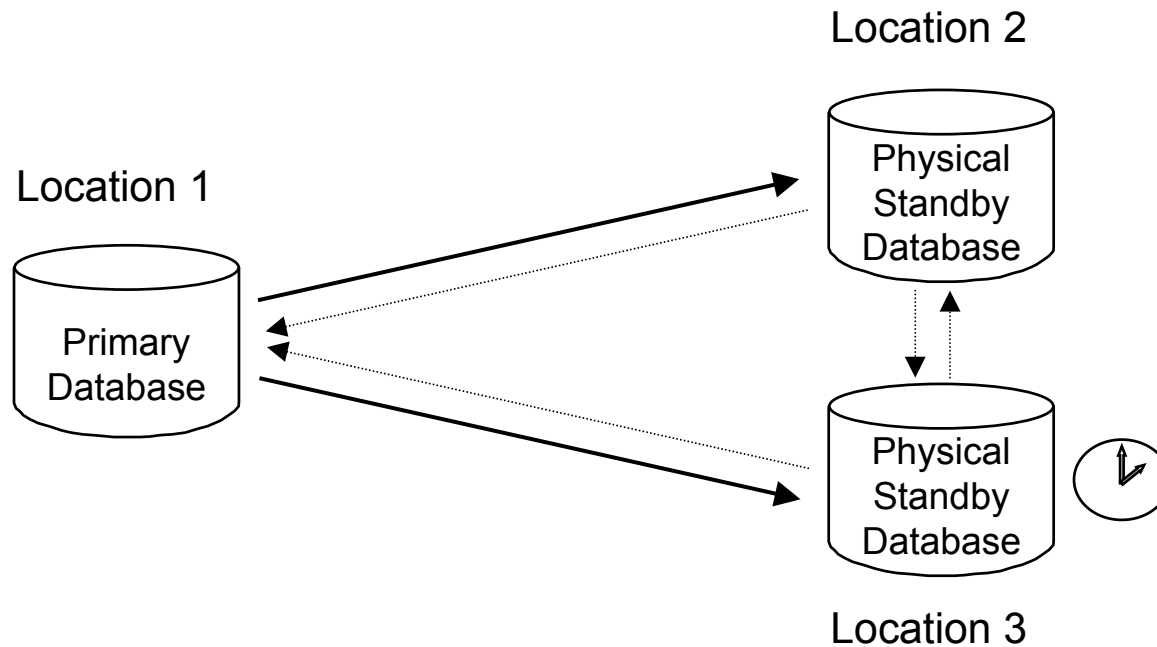


```
LOG_ARCHIVE_DEST_1='LOCATION=location1_directory'  
LOG_ARCHIVE_DEST_STATE_1=ENABLE  
LOG_ARCHIVE_DEST_2='SERVICE=location2'  
LOG_ARCHIVE_DEST_STATE_2=ENABLE
```

```
LOG_ARCHIVE_DEST_1='LOCATION=location2_directory'  
LOG_ARCHIVE_DEST_STATE_1=ENABLE  
LOG_ARCHIVE_DEST_2='SERVICE=location1'  
LOG_ARCHIVE_DEST_STATE_2=DEFER
```

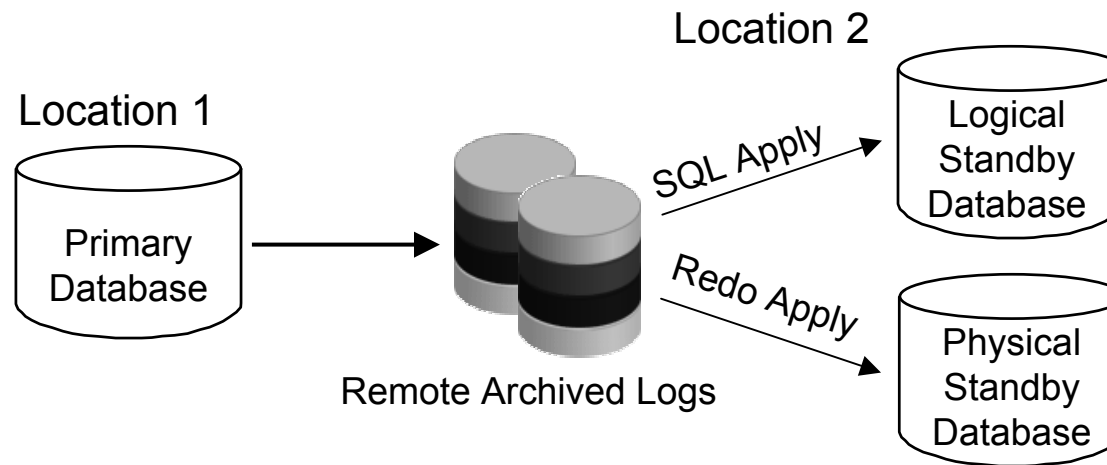
- One physical standby location provides basic disaster protection (a remote block-for-block copy of the primary database), but there is no additional protection in effect if either location fails
- Physical standby database can be used for reporting (redo apply must be temporarily paused)

Improved Physical Standby Configuration



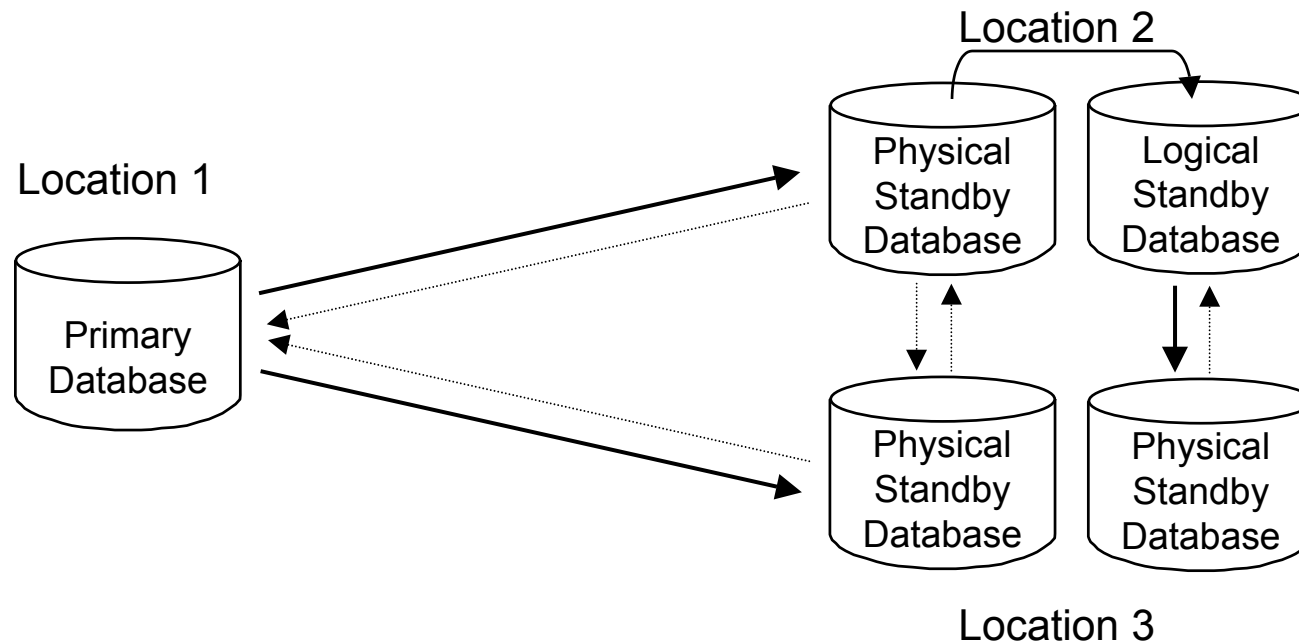
- Two physical standby locations maintain full disaster protection after any one location (primary or standby) fails
- One standby can be kept current with the primary database to facilitate fast failover while the other can be configured with a redo apply delay to create a “window of protection” against user error

Getting More From Your Standby Systems



- Physical standby (in recovery mode):
 - Maintains block-for-block copy of all primary data for disaster protection
 - Offloads database backups from primary
- Logical standby is optimized for continuous reporting, with additional:
 - Indexes
 - Materialized Views

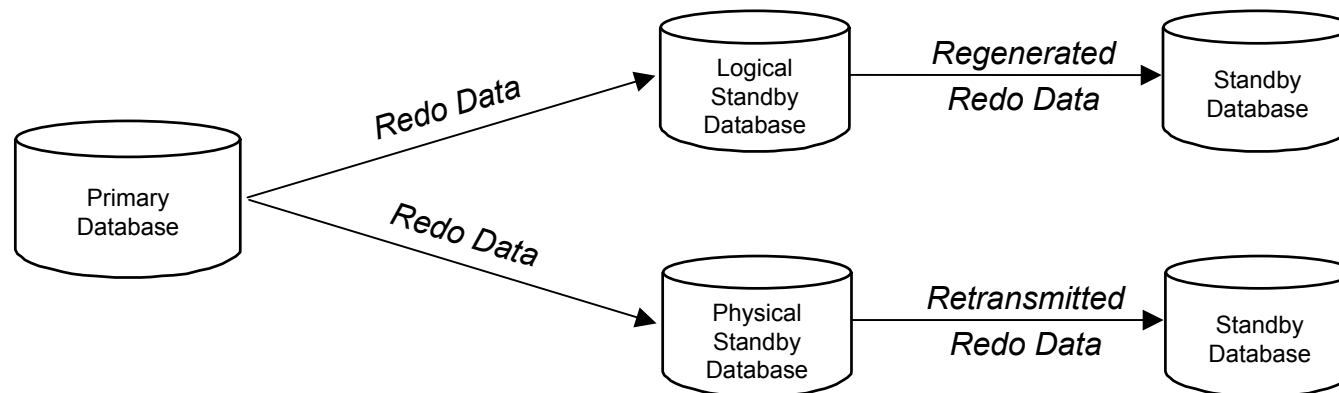
Getting More From Your Standby Systems (cont'd)



- Another physical standby can be used to provide disaster protection for the logical standby

Cascaded Redo Destinations

- Standby databases optionally can receive redo data from another standby database instead of the original primary database
- Primary database sends redo data only to selected standby databases and not to all standby databases
- Reduces the load on the primary system, and also reduces network traffic and use of valuable network resources around the primary site



Data Guard Resources

- Maximum Availability Architecture, best practices for Data Guard + RAC:
 - <http://otn.oracle.com/deploy/availability/htdocs/maa.htm>
- Data Guard page on OTN:
 - http://otn.oracle.com/deploy/availability/htdocs/dr_overview.html
- Oracle Internal HA Site
 - <http://oltp.us.oracle.com/>
- Oracle Internal Data Guard Site
 - <http://dr.us.oracle.com/>
- HA Mailing List
 - helpha_us@oracle.com
- Oracle EMEA Data Guard Cheat Sheet
 - http://files.oraclecorp.com/content/AllPublic/Workspaces/Marketing%20Sales%20Kits-Public/Oracle9i/Database/Reliability/ha_dg_cheatsheet.html
- Data Guard Consulting Accelerator
 - http://www.oracle.com/consulting/offerings/platform/dataguardaccel_ds.html

QUESTIONS

ANSWERS

ORACLE®