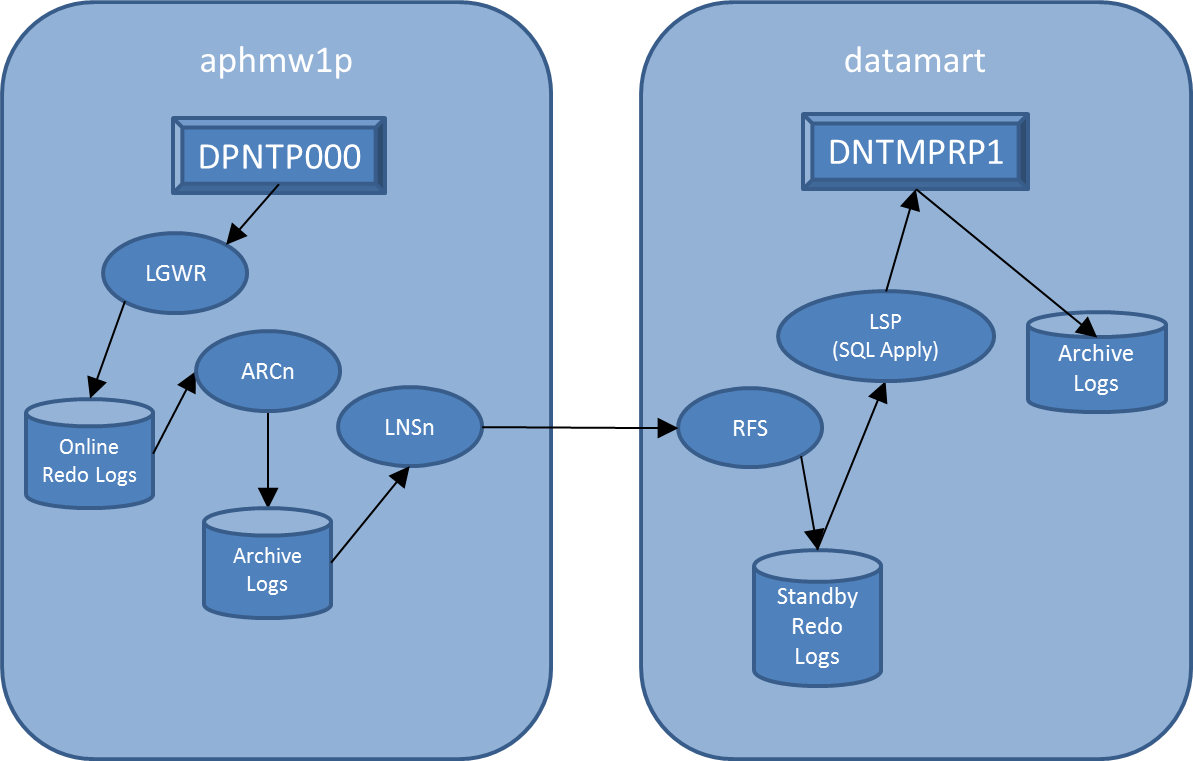
PIHMS Reporting Logical Standby Quick Reference

# Environment Overview

## Simplified Diagram

The diagram below is a simplified version of what is going on.



It doesn’t detail all of the Oracle processes involved in the logical standby. For more details on that see:

## Archivelogs

There are two types of archivelogs

* The archivelogs that are generated by the standby itself by the transactions being applied by the SQL Apply processes. These are stored in:  
  /oraarchivelog/pihmsu001/DNTMPRP1/archive
* The archivelogs that come from the primary, are shipped to the standby, and are mined to extract DDL and DML. These are stored in:   
  /oraarchivelog/pihmsu001/DNTMPRP1/stby\_archive

The archivelogs coming from the primary are maintained automatically by the log apply process and are currently configured to be deleted within one hour of their being successfully applied. The configuration is set by executing the following on the standby:

DBMS\_LOGSTDBY.APPLY\_SET(inname=>'LOG\_AUTO\_DEL\_RETENTION\_TARGET', value=>60);

The logs generated by the standby itself (/oraarchivelog/pihmsu001/DNTMPRP1/archive) are like regular archive log files. These are backed up and purged using regular rman archive log and database backups.

## Important configuration settings

On the primary:

|  |  |  |
| --- | --- | --- |
| Parameter | Setting | Comment |
| log\_archive\_dest\_state\_2 | ENABLE | Status of the remote archive destination for the standby. Can be changed to DEFER to halt log shipping. |
| log\_archive\_dest\_2 | SERVICE=DNTMPRP1.WORLD ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=DNTMPRP1 | Tells where, when and how to ship the logs to the standby. In this case it is asynchronously to DNTMPRP1 and only when the primary instance has the role of primary. |
| archive\_lag\_target | 900 | The maximum number of seconds between log switches. This controls the latency on the standby. |

On the standby:

|  |  |  |
| --- | --- | --- |
| Parameter | Setting | Comment |
| log\_archive\_dest\_1 | location=/oraarchivelog/pihmsu001/DNTMPRP1/archive VALID\_FOR=(ONLINE\_LOGFILES,ALL\_ROLES) DB\_UNIQUE\_NAME=DNTMPRP1' |  |
| log\_archive\_dest\_3 | location=/oraarchivelog/pihmsu001/DNTMPRP1/stby\_archive VALID\_FOR=(STANDBY\_LOGFILES,STANDBY\_ROLE) DB\_UNIQUE\_NAME=DNTMPRP1' |  |
| db\_file\_name\_convert |  |  |
| log\_file\_name\_convert |  |  |
| fal\_server | DPNTP000 |  |

## Documentation of interest

Dataguard   
<http://docs.oracle.com/cd/E11882_01/server.112/e41134/toc.htm>

DBMS\_LOGSTDBY Package  
<http://docs.oracle.com/cd/E11882_01/appdev.112/e40758/d_lsbydb.htm#ARPLS66827>

## PIHMS Customizations

PIHMS is customized to exclude the MYPIHMS and PIHMS\_EXPIMP schemas and ignore GRANT errors.

How do you set schema exclusion?

It's in the script - the customize standby part. Look in the $SCRIPTS/standby/sql directory under customize\_standby.sql

EXECUTE DBMS\_LOGSTDBY.SKIP('SCHEMA\_DDL','MYPIHMS', '%', NULL, TRUE, '\');

EXECUTE DBMS\_LOGSTDBY.SKIP('DML','MYPIHMS', '%', NULL, TRUE, '\');

so the 1st one says to skip DDL for MYPIHMS

the 2nd says skip and DML like insert/update/delete for MYPIHMS

Also found out we needed to skip PHIMS\_EXPIMP

because the datapump export creates temporary objects - some of which are unsupported.

They don't hurt anything, but they make errors in the logs.

# Procedures

## Suspend/Resume log transfer

If for some reason you need to stop log transfer to the standby database, run the following on the primary.

alter system set log\_archive\_dest\_state\_2=defer;

To resume log transfer, run:

alter system set log\_archive\_dest\_state\_2=enable;

## Resync the standby

Log onto the standby box (datarep) and go to the directory for the standby database.

cd $SCRIPTS/standby

First, you’ll need to edit the config file that contains credentials used for the resync.

vi configs/resync.parm

Change the values for the database credentials for both the primary and standby. They will look something like:  
  
DPNTP000\_USERNAME\_PW=somesysdba/mypass  
DNTMPRP1\_USERNAME\_PW=somesysdba/mypass

The credentials you enter here should be for an account that has SYSDBA permissions on both the primary and standby databases.

The other configuration items should be all set up.

Next, make sure that both the primary and standby databases are up and running. The standby needs to be running because the first step is to export the existing MYPIHMS schema prior to the duplication so that it can be reloaded again afterwards.

Now, you’re ready for the resync. To start, enter the following…

./resync\_standby.ksh

If an issue is encountered, read the logs indicated by the script output and resolve the issue. To restart the script where it left off, run:

./resync\_standby.ksh -r

If you need to restart the resync at a particular step (you can find the step numbers in the log file), you can do it by running:

./resync\_standby.ksh –t <step number>

When the script is done, the standby should be up and running. You can confirm by running the standby\_status.ksh script.

I am looking into your recync\_standby script

duplicate\_for\_standby function

Do you need to copy manually backupsets from primary server over to datamart

/orabackup/pihmsu000/DPNTP000/rman\_backups

Well... The answer in the long-term is no. For the short-term you do. I'm glad you brought it up because I should add it to the document. I had Mike Giordano set up TDP so that we didn't need to copy the backup over. It can just get it from the tape system by using a different tdpo.opt file. However when I tried using it the day I set up the standby, I kept getting an error. Found out it was because the version of TDP on datamart is older than the one on aphmw1p and the formats they use on the TDP server are incompatible so we can't do it directly until I get Giordano to update TDP.

But we're in the moratorium. So it won't be till January.

So for now we have to do a disk backup and copy the directory over.

I'll add that to the doc.

!!!! It utilizing tape now. !!!!

So you need to do scp backusets prior to execution for this resync\_standby script?

yes. I'll add it to the doc, but the added steps are: 1) take the disk backup on the primary (DPNTP000) and 2) scp over to datamart, 3) run the script

!!! No need above!!! Just take backup first

## Clear out un-shipped/applied logs on primary

If for some reason archivelog space is running low on the primary and there is a problem with getting the logs shipped to the standby, running the regular archivelog job might not clear out enough space because the deletion policy keeps logs until they are shipped to the standby. In this situation, it may be necessary to abandon the standby to keep the primary operational. To do this, it will be necessary to change the archivelog deletion policy on the primary and run a delete to clear the backed up logs that haven’t gone to the standby.

On the primary box, log on to rman with the primary as the target and change the configuration to:

CONFIGURE ARCHIVELOG DELETION POLICY TO NONE;

DELETE NOPROMPT ARCHIVELOG ALL BACKED UP 1 TIMES TO DEVICE TYPE sbt\_tape;

Once the issues are resolved, make sure to set it back to:

CONFIGURE ARCHIVELOG DELETION POLICY TO SHIPPED TO STANDBY;

## Check status of logical standby system

cd $SCRIPTS/standby

./standby\_status.ksh

## Skipping a Transaction

If for some reason there is a transaction that is causes SQL apply to crash and you need to skip it without reinitializing the standby, you can do so using DBMS\_LOGSTDBY.SKIP\_TRANSACTION and specifying the transaction id. The transaction id can be obtained from the logs or the standby\_status.ksh script output.

sqlplus / as sysdba  
SQL> EXECUTE DBMS\_LOGSTDBY.SKIP\_TRANSACTION (XIDUSN => 1, XIDSLT => 13, XIDSQN => 1726);

PL/SQL procedure successfully completed.

After skipping the transaction, try restarting the logical standby apply and re-checking the status:

sqlplus / as sysdba

SQL> ALTER DATABASE START LOGICAL STANDBY APPLY;

Database altered.

Then re-check the status. From the shell prompt:

cd $SCRIPTS/standby  
./standby\_status.ksh > stndby\_status\_1.out

!!! Was getting DNTMPRP1: Logical Standby Error emails

ndby\_state=SQL\_APPLY\_NOT\_ON : seconds\_latency=31753

Yes.  I used the steps in the Quick Reference for skipping a transaction.

First I ran the standby\_status.ksh script to see what was up.  I saw something like:

EVENT\_TIME          STATUS\_CODE STATUS                         EVENT\_PART                     XID

------------------- ----------- ------------------------------ ------------------------------ -------------------------

2014-01-02 07:57:11        1918 ORA-01918: user 'SYSMS' does   ALTER INDEX                    10.23.47547

                                not exist                      "PIHMS"."OTU\_ORG\_IDX" REBUILD

                                                               NOPARALLEL

2014-01-02 07:57:02       16222 ORA-16222: automatic Logical                                  ..

                                Standby retry of last

2014-01-02 07:57:02           0 Apply LWM 110278199, HWM                                      ..

                                110278199, SCN 110278241

2014-01-02 07:57:02       16111 ORA-16111: log mining and                                     ..

                                apply setting up

2014-01-02 07:56:59        1918 ORA-01918: user 'SYSMS' does   ALTER INDEX                    10.23.47547

                                not exist                      "PIHMS"."OTU\_ORG\_IDX" REBUILD

                                                               NOPARALLEL

2014-01-02 07:56:47       16111 ORA-16111: log mining and                                     ..

                                apply setting up

2014-01-02 07:56:47           0 Apply LWM 110278199, HWM                                      ..

                                110278199, SCN 110278241

I saw this once before following a build where they did something to modify the materialized views for the org hierarchy.  The failure is due to some bug where the username gets messed up in the replicated transaction.  You can see from the error message, it was complaining about the username SYSMS.  It’s an Oracle bug (13591624.8) that’s fixed in the next PSU.  You can see though that it was basically overwriting the first 3 characters of “PIHMS” (the right username) with SYS.

So since the failed transaction was just rebuilding the indexes and not actually manipulating data, I skipped those transactions by running:

EXECUTE DBMS\_LOGSTDBY.SKIP\_TRANSACTION(XIDUSN\_P => 10, XIDSLT\_P => 23, XIDSQN\_P => 47547)

-- Note, the numbers come from the XID in the status output for the failed transaction.

Then restarting the logical standby apply process…

sqlplus / as sysdba

alter database start logical standby apply;

Then checking the status again.  There was a total of 4 transactions (all index rebuilds) that I had to skip.  After I skipped the last one, it caught up in about 5 minutes.

One other thing. There's cron-job running to monitor the standby.

Hasn't alerted any yet

$SCRIPTS/standby/monitor\_standby.ksh

It monitors the latency.

If it gets above 30 min we get alerted

right now just e-mail