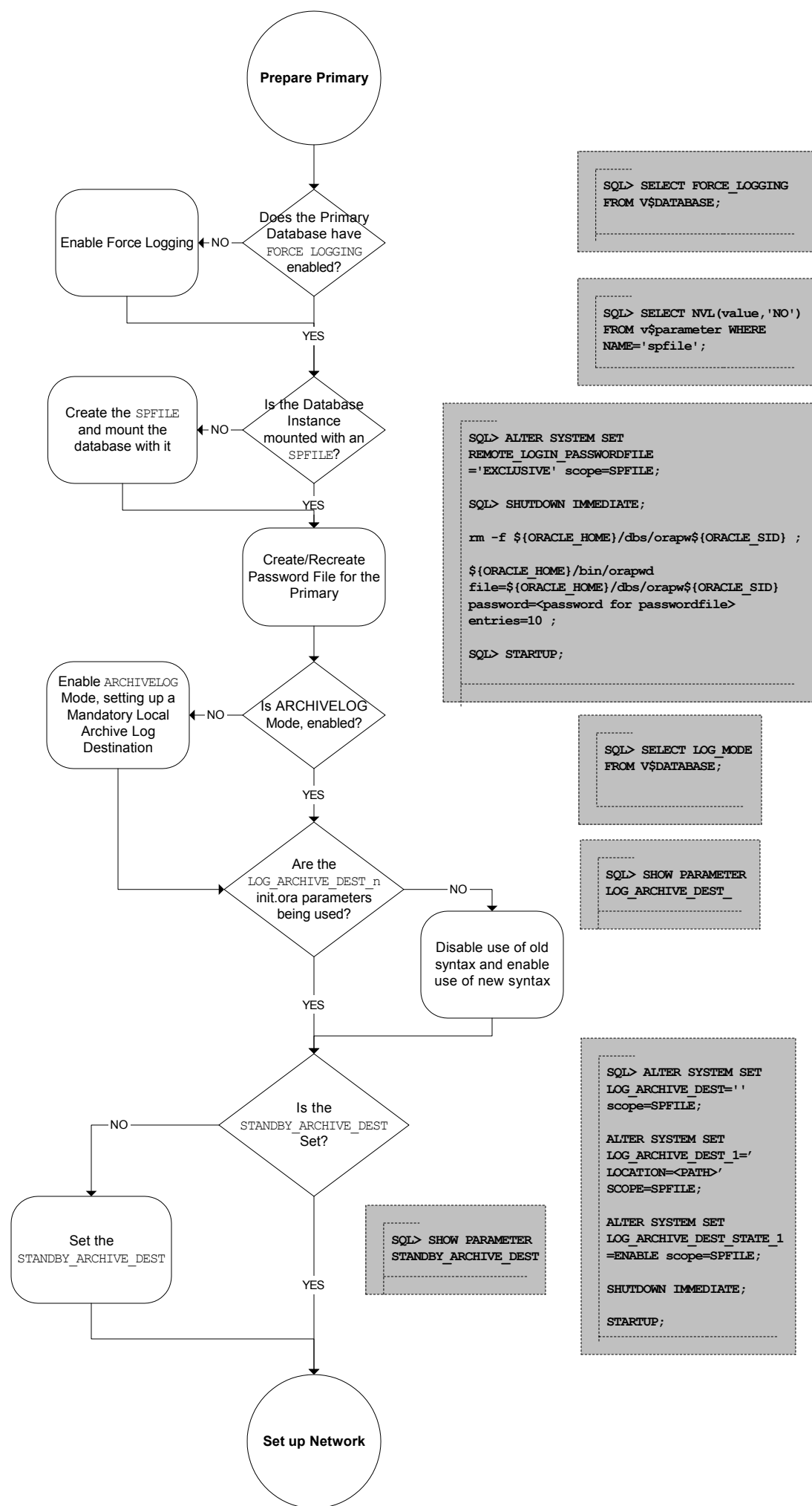


Prepare Primary



```
SQL> ALTER DATABASE FORCE LOGGING;
```

```
SQL> CREATE SPFILE FROM PFILE;
SHUTDOWN IMMEDIATE;
STARTUP;
```

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_1='LOCATION=<PATH>' SCOPE=BOTH;

ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_1=ENABLE SCOPE=BOTH;

ALTER SYSTEM SET LOG_ARCHIVE_FORMAT='<FORMAT>' SCOPE= SPFILE;

ALTER SYSTEM SET LOG_ARCHIVE_START=TRUE SCOPE=SPFILE;

ALTER SYSTEM SET DG_BROKER_START=TRUE SCOPE=BOTH;

SHUTDOWN IMMEDIATE;

STARTUP MOUNT;

ALTER DATABASE ARCHIVELOG;

ALTER DATABASE OPEN;

ARCHIVE LOG LIST;
```

```
SQL> ALTER SYSTEM SET STANDBY_ARCHIVE_DEST='<PATH>' SCOPE=BOTH;
```

```
SQL> SELECT FORCE_LOGGING FROM V$DATABASE;
```

```
SQL> SELECT NVL(value, 'NO') FROM v$parameter WHERE NAME='spfile';
```

```
SQL> ALTER SYSTEM SET REMOTE_LOGIN_PASSWORDFILE='EXCLUSIVE' scope=SPFILE;

SQL> SHUTDOWN IMMEDIATE;

rm -f ${ORACLE_HOME}/dbs/orapw${ORACLE_SID} ;

${ORACLE_HOME}/bin/orapwd file=${ORACLE_HOME}/dbs/orapw${ORACLE_SID} password=<password for passwordfile> entries=10 ;

SQL> STARTUP;
```

```
SQL> SELECT LOG_MODE FROM V$DATABASE;
```

```
SQL> SHOW PARAMETER LOG_ARCHIVE_DEST_
```

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST='' scope=SPFILE;

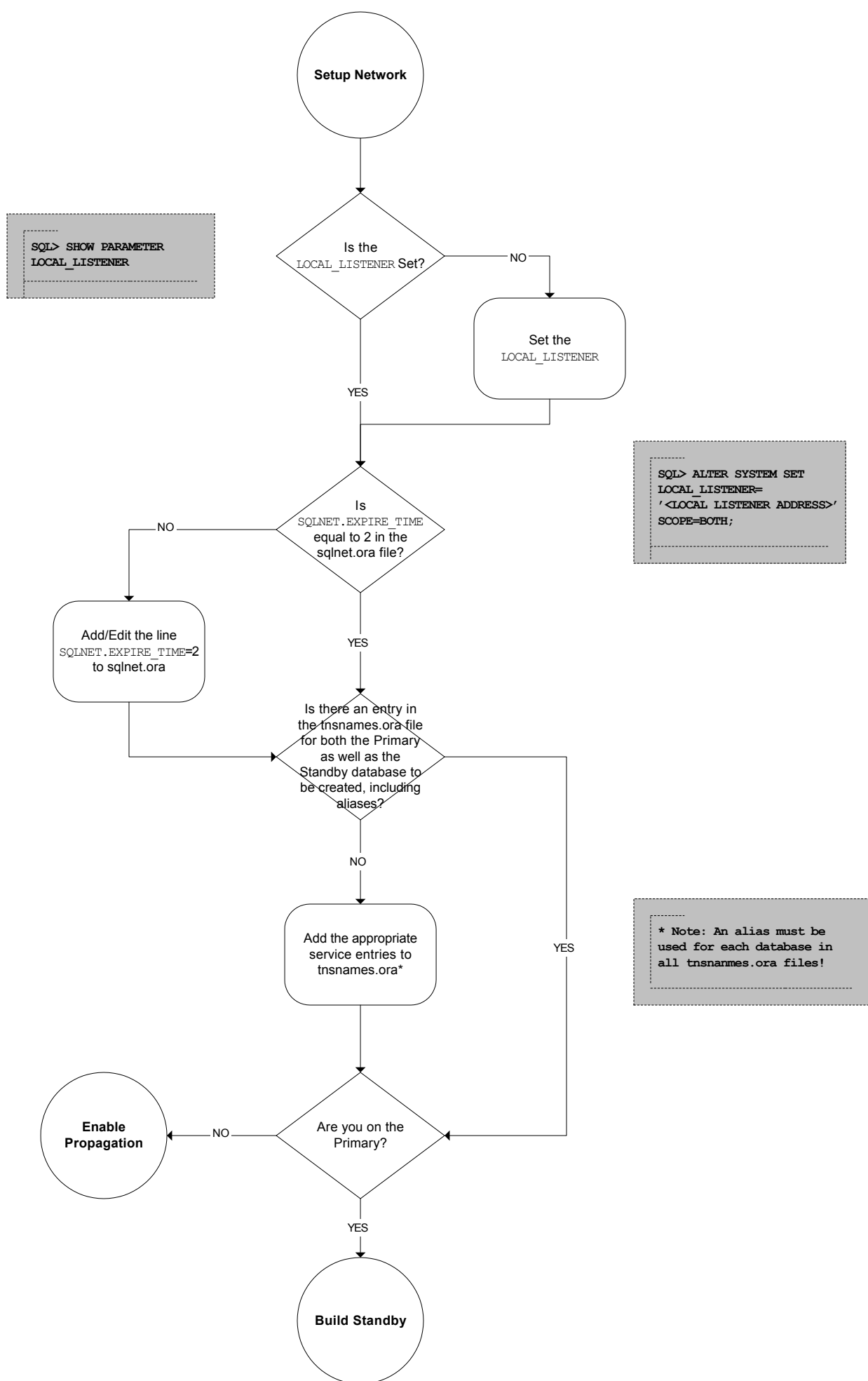
ALTER SYSTEM SET LOG_ARCHIVE_DEST_1='LOCATION=<PATH>' SCOPE=SPFILE;

ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_1=ENABLE scope=SPFILE;

SHUTDOWN IMMEDIATE;

STARTUP;
```

```
SQL> SHOW PARAMETER STANDBY_ARCHIVE_DEST
```



Setup Network

Is the LOCAL_LISTENER Set?

Set the LOCAL_LISTENER

YES

Is SQLNET.EXPIRE_TIME equal to 2 in the sqlnet.ora file?

NO

Add/Edit the line SQLNET.EXPIRE_TIME=2 to sqlnet.ora

YES

Is there an entry in the tnsnames.ora file for both the Primary as well as the Standby database to be created, including aliases?

NO

Add the appropriate service entries to tnsnames.ora*

Are you on the Primary?

NO

Enable Propagation

YES

Build Standby

SQL> ALTER SYSTEM SET LOCAL_LISTENER='<LOCAL LISTENER ADDRESS>' SCOPE=BOTH;

* Note: An alias must be used for each database in all tnsnames.ora files!

Build Standby

Take a Hot Backup of the Database per preferred method

Create Standby Controlfile (Note: This must have a distinct name from that of any existing controlfiles on the Primary)

```
SQL> ALTER DATABASE CREATE
STANDBY CONTROLFILE AS
'<PATH>';
```

Create a PFILE from SPFILE, which will become the basis for the Standby Instance's parameter file

```
SQL> CREATE PFILE=' PATH'
FROM SPFILE;
```

Create Directory Structures on the Standby Server

Package and Transfer Backupset to Standby Server (Datafiles; Standby Controlfile; and parameter file)

Unpack the Backup Set and move Datafiles/ Standby Controlfile/ Parameter File to desired locations

Edit the Parameter File on the Standby Host as appropriate (see box on left)

Create SPFILE from previously modified PFILE on the Standby Host

```
SQL> CREATE SPFILE
FROM PFILE=' <PATH>';
```

Create/Recreate Password File for the Primary

```
SQL> ALTER SYSTEM SET
REMOTE_LOGIN_PASSWORDFILE
='EXCLUSIVE' scope=SPFILE;

SQL> SHUTDOWN IMMEDIATE;

rm -f ${ORACLE_HOME}/dbs/orapw${ORACLE_SID} ;

${ORACLE_HOME}/bin/orapwd
file=${ORACLE_HOME}/dbs/orapw${ORACLE_SID}
password=<password for passwordfile>
entries=10 ;
```

Mount the Physical Standby and place in Managed Recovery mode (see box on left)

Setup Network

Edit/Add:

```
CONTROL_FILES='<PATH TO
STANDBY CONTROLFILE
YOU CREATED EARLIER>'

CORE_DUMP_DEST=<PATH>
USER_DUMP_DEST=<PATH>
BACKGROUND_DUMP_DEST=<PATH>
```

```
LOG_ARCHIVE_DEST_1=
'LOCATION=<PATH>'
```

```
AQ_TM_PROCESSES=0
```

```
STANDBY_ARCHIVE_DEST='<PATH>'
```

```
LOCAL_LISTENER=
'<LOCAL LISTENER ADDRESS>'
```

```
DG_BROKER_START=TRUE
```

For GAP Detection:

```
FAL_CLIENT=
'<LOCAL INSTANCE ALIAS ON
ON THE PRIMARY>'
```

```
FAL_SERVER=
'<PRIMARY SERVICE ALIAS>'
```

If Primary and Standby on the same host:

```
INSTANCE_NAME
LOCK_NAME_SPACE
```

If filesystem layouts differ between hosts:

```
DB_FILE_NAME_CONVERT
LOG_FILE_NAME_CONVERT
```

Optionally:

```
STANDBY_FILE_MANAGEMENT=
AUTO | MANUAL
```

(Note: Databases using raw file systems must use MANUAL)

```
SQL> STARTUP NOMOUNT;
```

```
ALTER DATABASE MOUNT
STANDBY DATABASE;
```

```
ALTER DATABASE RECOVER
MANAGED STANDBY DATABASE
DISCONNECT FROM SESSION;
```

```
# SELECT DATABASE ROLE
# FROM V$DATABASE;
# should yield 'PHYSICAL
# 'STANDBY'
```

Enable Propagation

Set initialization parameters on Primary to initiate the propagation of archive logs to the Standby

Start Remote Archiving by generating an Archive Log

Would you like to use the Data Guard Broker to manage the configuration?

Data Broker Setup

Verify Standby (Manual)

END
(Physical Standby Created)

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_2='SERVICE=<SERVICE NAME> {OPTIONS}' SCOPE=BOTH;  
  
ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=ENABLE SCOPE=BOTH;  
  
ALTER SYSTEM SET FAL_CLIENT='' SCOPE=BOTH;  
  
ALTER SYSTEM SET FAL_SERVER='' SCOPE=BOTH;  
  
ALTER SYSTEM SET AQ_TM_PROCESSES=1 SCOPE=BOTH;
```

```
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;  
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

YES

NO

```

. oraenv ${ORACLE_SID}
sqlplus "/ as sysdba"
SQL> shutdown immediate;
SQL> startup nomount;
SQL> alter system set
instance_name='<new instance name>' scope=SPFILE;
SQL> alter system set
service_names='<new instance name>.db_domain' scope=SPFILE;
SQL> shutdown immediate;

rm ${ORACLE_HOME}/dbs/orapw${ORACLE_SID};
rm ${ORACLE_HOME}/dbs/init${ORACLE_SID}.ora;
${ORACLE_HOME}/bin/orapwd
file=${ORACLE_HOME}/dbs/orapw${ORACLE_SID} password=<pwd>;
mv spfile${ORACLE_SID}.ora spfile<new instance name>.ora
vi /etc/oratab (replace ${ORACLE_SID} with new instance name)

. oraenv <new instance name>
sqlplus "/ as sysdba"
SQL> startup nomount;
SQL> alter database mount standby database;
SQL> alter database recover managed standby
database disconnect from session;

```

```

DGMGRL>
connect sys/<pwd from pwdfile>@"<primary service alias>";

CREATE CONFIGURATION '<config name>' AS
PRIMARY SITE IS '<primary site name>'
RESOURCE IS '<primary resource name>'
HOSTNAME IS '<hostname from v$instance>'
INSTANCE NAME IS '<instance name from v$instance>'
SERVICE NAME IS '<primary service alias>'
SITE IS MAINTAINED AS PHYSICAL;

CREATE SITE '<standby site name>'
RESOURCE IS '<standby resource name>'
HOSTNAME IS '<hostname from v$instance>'
INSTANCE NAME IS '<instance name from v$instance>'
SERVICE NAME IS '<standby service alias>'
SITE IS MAINTAINED AS PHYSICAL;

ENABLE SITE '<primary site name>';
ENABLE SITE '<standby site name>';
ENABLE CONFIGURATION;

```

```

DGMGRL>
connect sys/<pwd from pwdfile>@"<primary service alias>";

ALTER RESOURCE '<standby resource name>'
set property StandbyFileManagement = 'AUTO';

ALTER RESOURCE '<standby resource name>'
set property StandbyArchiveDest =
'LOCATION=<STANDBY_ARCHIVE_DEST>';

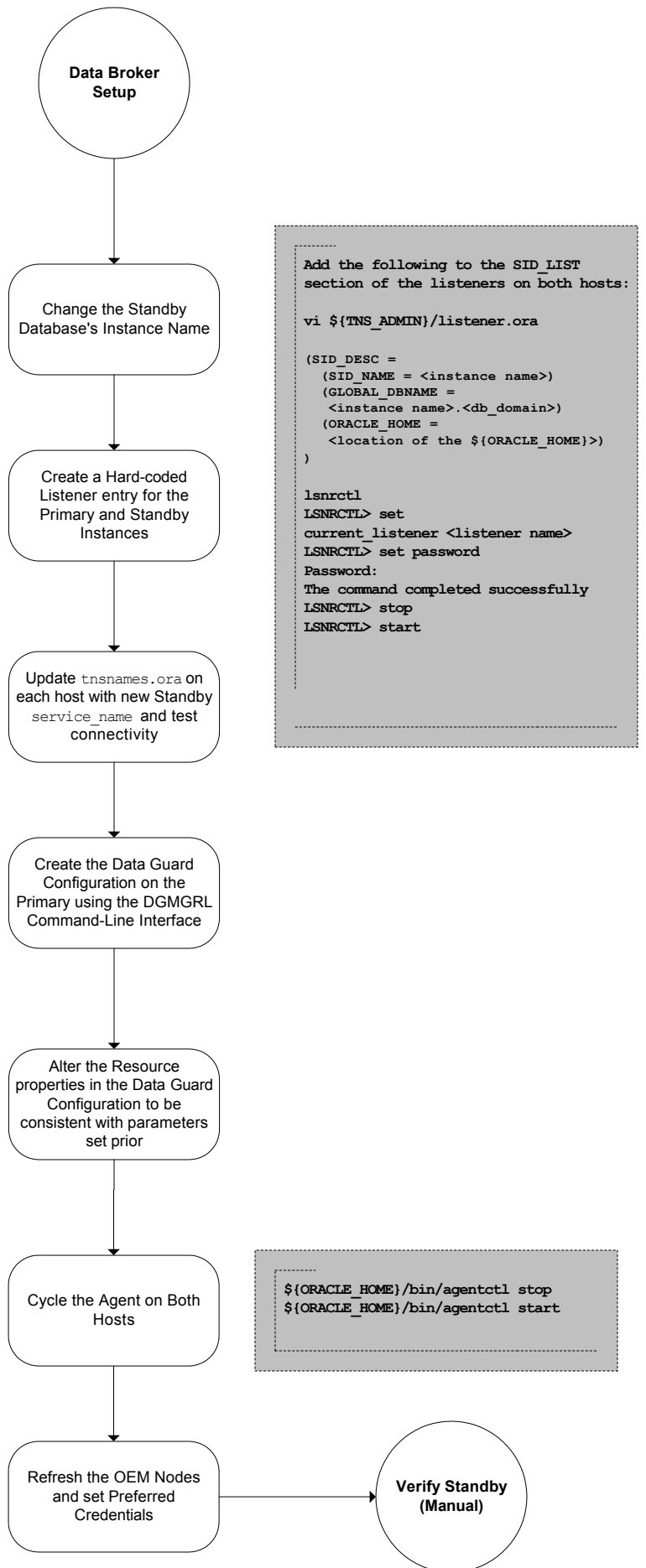
ALTER RESOURCE '<standby resource name>'
set property LogArchiveFormat = '${ORACLE_SID}_%t_%S
.arc';

ALTER RESOURCE '<standby resource name>'
set property DbFileNameConvert = '<path>, <replace path>';

ALTER RESOURCE '<standby resource name>'
set property LogFileNameConvert = '<path>, <replace path>';

ALTER RESOURCE '<primary resource name>'
set property StandbyFileManagement = 'AUTO';

```



Add the following to the SID_LIST section of the listeners on both hosts:

```

vi ${TNS_ADMIN}/listener.ora

(SID_DESC =
(SID_NAME = <instance name>)
(GLOBAL_DBNAME =
<instance name>.<db_domain>)
(ORACLE_HOME =
<location of the ${ORACLE_HOME}>)
)

lsnrctl
LSNRCTL> set
current_listener <listener name>
LSNRCTL> set password
Password:
The command completed successfully
LSNRCTL> stop
LSNRCTL> start

```

```

${ORACLE_HOME}/bin/agentctl stop
${ORACLE_HOME}/bin/agentctl start

```

```
SQL> SELECT DEST_NAME,
STATUS,
TYPE
FROM V$ARCHIVE_DEST_STATUS
WHERE TYPE='LOGICAL';
```

Verify Standby
(Manual)

Does the Primary show that the Standby has a status of 'VALID' for its archive log destination?

Verify SQL*Net configuration and make necessary modifications on both Primary and Standby hosts via Isnrctl, tnsping, etc; Double-check init.ora parameters on Primary Host set during Enable Archiving to Standby sub-process

```
# RUN ON THE STANDBY
SELECT COUNT(*)
FROM V$MANAGED_STANDBY
WHERE PROCESS='MRP0';
```

Is the Managed Standby Recovery Process running? (a.k.a. returns 1)

Bounce Managed Recovery Process

```
SQL> ALTER DATABASE
RECOVER MANAGED STANDBY
DATABASE CANCEL;

SHUTDOWN IMMEDIATE;
STARTUP NOMOUNT;

ALTER DATABASE MOUNT
STANDBY DATABASE;

ALTER DATABASE RECOVER
MANAGED STANDBY DATABASE
DISCONNECT FROM SESSION;
```

```
# RUN ON THE PRIMARY
SELECT MAX(AL.SEQUENCE#) SEQUENCE,
AL.ARCHIVED,
AL.DEST_ID,
AD.DEST_NAME,
AL.STANDBY_DEST
FROM V$ARCHIVE_DEST AD,
V$ARCHIVED_LOG AL
WHERE AL.DEST_ID = AD.DEST_ID
AND AL.STANDBY_DEST='YES'
AND AL.ARCHIVED = 'YES'
GROUP BY AL.ARCHIVED,
AL.DEST_ID,
AD.DEST_NAME,
AL.STANDBY_DEST;
```

Verify that Archive logs are being shipped to the Standby on the Primary; Annotate Max Archived Sequence #

Verify that the Standby Database is receiving and applying; compare results to previous query; Annotate Max Applied Sequence #

```
# RUN ON THE STANDBY
SELECT MAX(AL.SEQUENCE#) SEQUENCE,
FROM V$ARCHIVED_LOG AL
WHERE AL.DEST_ID=<ID FROM ABOVE>
AND AL.APPLIED='YES';
```

END
(Physical Standby Operating Successfully)

Is the Max Archived Sequence # equal to Max Applied Sequence #?

Retry (NO)

```
# RUN ON THE STANDBY
# FOR EACH COPIED
# ARCH LOG
SQL> ALTER DATABASE
REGISTER PHYSICAL LOGFILE
'FULLPATH TO FILE';
```

For the LOW_SEQUENCE# through the HIGH_SEQUENCE# for each gap entry, copy the archive logs manually to the Standby and register manually

Is the discrepancy due to an Archive Log Gap? (Yes, if results are returned)

```
# RUN ON THE STANDBY
SELECT *
FROM V$ARCHIVE_GAP;
```

