

Creating a Physical Standby Database – Scripted Example (Unix)

James L. Colestock

The following are step-by-step instructions on how to build a Physical Data Guard configuration. There are two accompanying tar files (one for the primary database server; one for the standby database server), which contain the scripts that simply need to be edited and run in order to build the configuration in question. Such scripts are necessary, in large part, due to the failings of Oracle's Data Guard Broker GUI, which has never successfully been able to create Standby Databases via its wizard – more recent releases of the OEM tool in question are known to work much more reliably. Once a Standby Database is created, however, the Data Guard Broker GUI can be used to manage the configuration.

Assumptions:

- The Primary and Standby databases will reside on separate physical hosts; therefore, we won't specify the `lock_space_name` `init.ora` parameter
- Oracle 9i Release 2 databases/binaries are installed/used on both the Primary and Standby Hosts
- The Primary database already exists
- The `ARCn/RFS` background processes will be used to propagate changes - in the form of archive logs - from Primary to Standby; in other words, the 'Maximum Performance' configuration is employed in this example
- Listeners already exist on the Primary and Standby hosts
- The `ORACLE_HOME` – in terms of location and version – are identical on each host
- The databases use the same `db_domain`
- There are substantive differences in the file systems between the Primary and Standby hosts
- HP-UX was the platform used for this sample
- Korn shell is the default `oracle` user shell
- `/usr/bin/ksh` is the location of the Korn shell executable

On the Primary database server:

Create a directory and extract the Data Guard example scripts into it:

```
oracle@primary> mkdir -p ${ORACLE_BASE}/build_dataguard
oracle@primary> cd ${ORACLE_BASE}/build_dataguard
oracle@primary> tar -tvf dg_scripts_primary.tar
rwx----- 106/101    2079 Jan 26 15:11 2006 create_dg_config.ksh
rwx----- 106/101    1481 Jan 26 14:33 2006 enable_config.ksh
rwx----- 106/101    3143 Jan 17 13:38 2006 hot_backup.ksh
rwx----- 106/101    2959 Jan 26 11:00 2006 prep_primary.ksh
rwx----- 106/101    3602 Jan 26 13:57 2006 setup_net.ksh
rwx----- 106/101     608 Jan 25 10:02 2006 transfer_backup.ksh
oracle@primary> tar -xvf dg_scripts_primary.tar
oracle@primary> chown oracle:dba *
oracle@primary> chmod 700 *
```

Using [Appendix A](#) as a reference, change the following values to suit your environment in `prep_primary.ksh`:

PRIMARY_HOST, PRIMARY_LSNR_PORT, LOCAL_LOG_ARCHIVE_DEST, STANDBY_ARCHIVE_DEST, SCRIPT_DIR, PWDFILE_PWD

```
oracle@primary> vi prep_primary.ksh
```

Using [Appendix A](#) as a reference, change the following values to suit your environment in `hot_backup.ksh`:

SCRIPT_DIR, LOCAL_BACKUP_DIR, LINK_TO_BACKUP_DIR, BACKUP_SET

```
oracle@primary> vi hot_backup.ksh
```

Using [Appendix A](#) as a reference, change the following values to suit your environment in `setup_net.ksh`:

TNS_ADMIN, PRIMARY_SERVICE, PRIMARY_HOST, PRIMARY_LSNR_PORT, STANDBY_SERVICE, STANDBY_SERVICE_ALIAS, STANDBY_HOST, STANDBY_LSNR_PORT

```
oracle@primary> vi setup_net.ksh
```

Set and verify your environment and then run the appropriate scripts. The `prep_primary.ksh` script prepares the database in question for entering the Data Guard configuration; `setup_net.ksh` modifies Oracle's networking files as appropriate; and `hot_backup.ksh` creates a backup set -- which will be transferred to the standby site in order to create the standby -- consisting of a standby control file, parameter file, and gzipped datafiles. Run the following, using your `${ORACLE_SID}`:

```
oracle@primary> . oraenv jlcdb
oracle@primary:jlcdb> echo $ORACLE_SID
jlcdb
oracle@primary:jlcdb> ./prep_primary.ksh
oracle@primary:jlcdb> ./setup_net.ksh
oracle@primary:jlcdb> ./hot_backup.ksh
```

On the Standby database server:

Create a directory and extract the Data Guard example scripts into it:

```
oracle@standby> mkdir -p ${ORACLE_BASE}/build_dataguard
oracle@standby> cd ${ORACLE_BASE}/build_dataguard
oracle@standby> tar -tvf dg_scripts_standby.tar
rwx----- 106/101    2634 Jan 25 18:05 2006 alter_pfile.ksh
rwx----- 106/101    2623 Jan 26 19:15 2006 build_standby.ksh
rwx----- 106/101    1410 Jan 18 17:11 2006 cr8_stdby_dirs.ksh
rwx----- 106/101    1482 Jan 26 22:33 2006 enable_config.ksh
rwx----- 106/101    3602 Jan 26 22:06 2006 setup_net.ksh
oracle@standby> tar -xvf dg_scripts_standby.tar
oracle@standby> chown oracle:dba *
oracle@standby> chmod 700 *
```

Using [Appendix A](#) as a reference, change the following values to suit your environment in `cr8_stdby_dirs.ksh`:

ORACLE_SID, ORACLE_BASE, SCRIPT_DIR, LOCAL_BACKUP_DIR, LINK_TO_BACKUP_DIR

Alter this script as appropriate to accommodate your desired directory layout as well.

```
oracle@standby> vi cr8_stdby_dirs.ksh
```

The `cr8_stdby_dirs.ksh` script will create the directories necessary for the standby database build. Run the following:

```
oracle@standby> ./cr8_stdby_dirs.ksh
```

On the Primary database server:

Using [Appendix A](#) as a reference, change the following values to suit your environment in `transfer_backup.ksh`:

SCRIPT_DIR, LINK_TO_BACKUP_DIR, BACKUP_SET, STANDBY_HOST

```
oracle@standby> vi transfer_backup.ksh
```

Set and verify your environment and then run the `transfer_backup.ksh` to transfer the backupset to the standby database server. Provide the password for the oracle user on the `${STANDBY_HOST}` when prompted:

```
oracle@primary> . oraenv jlcdb
oracle@primary:jlcdb> echo $ORACLE_SID
jlcdb
oracle@primary:jlcdb> ./transfer_backup.ksh
You will be prompted for oracle's password on standby.colestock.com...
Password:
jlcdb.tar                                100%   34MB   17.2MB/s   00:02
```

On the Standby database server:

Using [Appendix A](#) as a reference, change the following values to suit your environment in `build_standby.ksh`:

ORACLE_SID, SCRIPT_DIR, LINK_TO_BACKUP_DIR, BACKUP_SET, DATAFILE_LOC, BCF_LOC, PFILE_LOC, ORACLE_HOME, ORATAB_LOC, PWDFILE_PWD

```
oracle@standby> vi build_standby.ksh
```

Note that in this example that there is only a single datafile location being used. Once the standby database is built, you can move and rename datafiles, alter `*file_convert` parameters etc. at your discretion. Reference Oracle Data Guard Concepts and Administration Guide; Part No. A96653-02.

Using [Appendix A](#) as a reference, change the following values to suit your environment in `alter_pfile.ksh`:

ORACLE_SID, SCRIPT_DIR, ADMIN_DIR_BASE, BCF_LOC, STANDBY_SERVICE, ORACLE_HOME, LOCAL_LOG_ARCHIVE_DEST, STANDBY_ARCHIVE_DEST, STANDBY_HOST, STANDBY_LSNR_PORT

The following lines in the script also need to be modified:

```
echo "*.db_file_name_convert=('/ora1/', '/ots/oracle/', '/ora2/', '/ots/oracle/',
'/ora3/', '/ots/oracle/')" >> /var/tmp/init${ORACLE_SID}.tmp.ora

echo "*.log_file_name_convert=('/ora1/', '/ots/oracle/', '/ora2/', '/ots/oracle/',
'/ora3/', '/ots/oracle/')" >> /var/tmp/init${ORACLE_SID}.tmp.ora
```

Alter the above with attribute pairs that perform the conversions that are appropriate for your environment.

```
oracle@standby> vi alter_pfile.ksh
```

Run the following to build the standby database from the backupset previously transferred. The `build_standby.ksh` script calls `alter_pfile.ksh`. The script creates the standby database; sets all appropriate parameters; and mounts it in managed recovery mode:

```
oracle@standby> ./build_standby.ksh
```

Verify the Standby build by spot-checking the standby database's alert log; a successful build should yield the following:

```
oracle@standby> . oraenv jlcdb
oracle@standby:jlcdb> echo ${ORACLE_SID}
jlcdb
oracle@standby:jlcdb> tail ${ORACLE_BASE}/admin/${ORACLE_SID}/bdump/alert_${ORACLE_SID}.log
Standby Database mounted.
Completed:  alter database mount standby database
Wed Jan 18 10:34:20 2006
  alter database recover managed standby database disconnect from session
Attempt to start background Managed Standby Recovery process
MRP0 started with pid=13
MRP0: Background Managed Standby Recovery process started
Media Recovery Waiting for thread 1 seq# 66
Wed Jan 18 10:34:25 2006
Completed:  alter database recover managed standby database
```

Just as you performed on the primary database server and using [Appendix A](#) as a reference, change the following values to suit your environment in `setup_net.ksh`:

**TNS_ADMIN, PRIMARY_SERVICE, PRIMARY_SERVICE_ALIAS, PRIMARY_HOST,
PRIMARY_LISTENER_PORT, STANDBY_SERVICE, STANDBY_HOST, STANDBY_LISTENER_PORT**

```
oracle@standby> vi setup_net.ksh
```

Run the script to configure the oracle network for the Data Guard configuration:

```
oracle@standby> . oraenv jlcdb
oracle@standby:jlcdB> echo ${ORACLE_SID}
jlcdB
oracle@standby:jlcdB> ./setup_net.ksh
```

ON BOTH the Primary and Standby database servers:

Using [Appendix A](#) as a reference, change the following values to suit your environment in `enable_config.ksh`:

STANDBY_SERVICE_ALIAS, PRIMARY_SERVICE_ALIAS

```
oracle@primary> vi enable_config.ksh
oracle@standby> vi enable_config.ksh
```

Run the script to “enable” the Data Guard configuration, initiating the propagation and subsequent application of archive logs from the primary to the standby database server:

```
oracle@primary> . oraenv jlcdb
oracle@primary:jlcdB> echo ${ORACLE_SID}
jlcdB
oracle@primary:jlcdB> ./enable_config.ksh
```

```
oracle@standby> . oraenv jlcdb
oracle@standby:jlcdB> echo ${ORACLE_SID}
jlcdB
oracle@standby:jlcdB> ./enable_config.ksh
```

At this point, you should have a fully-functioning Physical Data Guard configuration!

Data Guard Broker Set-up (Optional)

On the Standby database server:

Change the Instance name of the Standby Database to facilitate management through OEM, etc.

Source the Standby instance; Change the Instance Name (i.e. jlcdb to jlcdb); Update all dependent files; and remount the Standby Database. Example:

Source the Database

```
oracle@standby> . oraenv jlcdb
oracle@standby:jlcdbs> echo ${ORACLE_SID}
jlcdbs
```

Change the Instance Name

```
oracle@standby:jlcdbs> sqlplus "/ as sysdba"
SQL> shutdown immediate;
SQL> startup nomount;
SQL> alter system set instance_name='jlcdb' scope=SPFILE;
SQL> alter system set service_names='jlcdb.colestock.com' scope=SPFILE;
SQL> shutdown immediate;
SQL> exit;
```

Update dependent files

```
oracle@standby:jlcdbs> rm ${ORACLE_HOME}/dbs/orapwjlcdb;
oracle@standby:jlcdbs> rm ${ORACLE_HOME}/dbs/initjlcdbs.ora;
oracle@standby:jlcdbs> $ORACLE_HOME/bin/orapwd file=${ORACLE_HOME}/dbs/orapwjlcdb
password=manager;
oracle@standby:jlcdbs> mv spfilejlcdbs.ora spfilejlcdb.ora
oracle@standby:jlcdbs> vi /etc/oratab (I replaced jlcdb with jlcdb)
```

Source the new \${ORACLE_SID} and remount the Standby database

```
oracle@standby> . oraenv jlcdb
oracle@standby:jlcdb> echo ${ORACLE_SID}
jlcdb
oracle@standby:jlcdb> sqlplus "/ as sysdba"
SQL> startup nomount;
SQL> alter database mount standby database;
SQL> alter database recover managed standby database disconnect from session;
SQL> exit;
```

ON BOTH the Primary and Standby database servers:

Create a hard-coded listener entry for each instance:

Add the following to the SID_LIST (make sure the SID_NAME is the name of the instance that resides on the host; GLOBAL_DBNAME should be the corresponding service_name entry):

On the Standby:

```
oracle@standby> vi ${TNS_ADMIN}/listener.ora

(SID_DESC =
  (SID_NAME = jlcdb)
  (GLOBAL_DBNAME = jlcdb.colestock.com)
  (ORACLE_HOME = /opt/oracle/product/9.2.0.6)
)

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

On the Primary:

```
oracle@primary> vi ${TNS_ADMIN}/listener.ora

(SID_DESC =
  (SID_NAME = jlcdb)
  (GLOBAL_DBNAME = jlcdb.colestock.com)
  (ORACLE_HOME = /opt/oracle/product/9.2.0.6)
)

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

Update the \${TNS_ADMIN}/tnsnames.ora file on both hosts to reflect the changes to the Standby and retest connectivity – specifically replace 'service_name=<service_name>' with the Standby's new service_name set above.

On the Primary database server*:

Using [Appendix A](#) as a reference, change the following values to suit your environment in create_dg_config.ksh:

**PRIMARY_HOST, PRIMARY_HOSTNAME, STANDBY_HOST, STANDBY_HOSTNAME,
PWDFILE_PWD, PRIMARY_SERVICE, STANDBY_SERVICE, STANDBY_SERVICE_ALIAS
PRIMARY_SERVICE_ALIAS, STANDBY_ARCHIVE_DEST**

The following lines in the script also have to be modified:

```
ALTER RESOURCE 'Standby_DB' set property DbFileNameConvert = '/ora1/, /ots/oracle/  
e/, /ora2/, /ots/oracle/, /ora3/, /ots/oracle/';
```

```
ALTER RESOURCE 'Standby_DB' set property LogFileNameConvert = '/ora1/, /ots/oracle/  
le/, /ora2/, /ots/oracle/, /ora3/, /ots/oracle/';
```

Alter the above with attribute pairs that perform the conversions that are appropriate for your environment.

```
oracle@primary> vi create_dg_config.ksh
```

Run the following to create the Data Guard Broker configuration via the DGMGRL Command-Line Utility:

```
oracle@primary> ./create_dg_config.ksh
```

*** Alternatively, you may perform this task from the Data Guard Broker GUI at the end of this document, instead of creating the configuration via DGMGRL the Command-line interface**

ON BOTH the Primary and Standby database servers:

Cycle the Intelligent Agents to refresh the hosts' configuration files:

```
oracle@standby> ${ORACLE_HOME}/bin/agentctl stop  
oracle@standby> ${ORACLE_HOME}/bin/agentctl start
```

```
oracle@primary> ${ORACLE_HOME}/bin/agentctl stop  
oracle@primary> ${ORACLE_HOME}/bin/agentctl start
```

As sysman, login to the OEM Repository and refresh the Primary and Standby nodes; set Preferred Credentials; and Grant Access to Targets as appropriate.

Navigate to Tools > Database Applications > Data Guard Manager. If you created a configuration you should see it now (Provide the sys/<pwd from pwdfile> for the primary database and you should see all the sites in your configuration); otherwise, create one now by following the wizard.

At this point, you should have a fully-functioning Data Guard Broker configuration to support your Physical Data Guard configuration!

Appendix A

Environment Variable	Example
PRIMARY_HOST =<fully-qualified hostname of primary> ;	<i>primary.colestock.com</i>
PRIMARY_LISTENER_PORT =<listener port assigned to the primary database> ;	<i>1521</i>
LOCAL_LOG_ARCHIVE_DEST =<local arch log dest > ;	<i>/ora1/oraarch/\${ORACLE_SID}</i>
STANDBY_ARCHIVE_DEST =<local arch dest for remote logs> ;	<i>\${LOCAL_LOG_ARCHIVE_DEST}/stdby</i> <i>(This directory is used when the Primary database takes on the Physical Standby database role)</i>
SCRIPT_DIR =<directory location of the extracted scripts> ;	<i>\${ORACLE_BASE}/build_dataguard</i>
LOCAL_BACKUP_DIR =<directory location for backup> ;	<i>/ora1/oradata/dataguard_build/\${ORACLE_SID}</i>
LINK_TO_BACKUP_DIR =<symbolic link to LOCAL_BACKUP_DIR location> ;	<i>\${SCRIPT_DIR}/backups</i> <i>(This directive is necessary to account for any differences in the file system layouts between the Primary and Standby database hosts. Therefore, this location should be available on both hosts)</i>
BACKUP_SET =<location of the backupset tar file>;	<i>\${LINK_TO_BACKUP_DIR}/\${ORACLE_SID}.tar ;</i> <i>(The backupset will contain: gzipped datafiles; standby controlfile; copy of the primary's pfile)</i>
TNS_ADMIN =<location of oracle networking files> ;	<i>\${ORACLE_HOME}/network/admin</i>
PRIMARY_SERVICE =<service name of the primary database>;	<i>mydb.colestock.com</i> <i>(Normally, instance_name.db_domain)</i>
PRIMARY_SERVICE_ALIAS =<alias used to refer to the primary service>;	<i>mydbp.colestock.com</i> <i>(This will be placed in tnsnames.ora and used by the standby to refer to the primary)</i>
STANDBY_SERVICE =<service name of the standby database>;	<i>mydb.colestock.com</i> <i>(Normally, instance_name.db_domain)</i>
STANDBY_SERVICE_ALIAS =<alias used to refer to the standby service>;	<i>mydbs.colestock.com</i> <i>(This will be placed in tnsnames.ora and used by the primary to refer to the standby)</i>
STANDBY_HOST =<fully-qualified hostname of standby>;	<i>standby.colestock.com</i>
STANDBY_LISTENER_PORT =<listener port assigned to the standby database> ;	<i>1521</i>
ORACLE_SID =<instance/db_name of the database>;	<i>mydb</i>
ORACLE_BASE =<oracle software base>;	<i>/ots/oracle</i>
ORACLE_HOME =<oracle binary location for the database>;	<i>/ots/oracle/product/9.2.0.6</i>
DATAFILE_LOC =<location for the datafiles on the standby>;	<i>/ots/oracle/oradata/\${ORACLE_SID}</i>
BCF_LOC =<location for standby controlfile on the standby>;	<i>/ots/oracle/oradata/\${ORACLE_SID}</i>
PFILE_LOC =<location for the parameter file on the standby>;	<i>/ots/oracle/admin/\${ORACLE_SID}/pfile</i>
ADMIN_DIR_BASE =<base directory for admin directories>;	<i>/ots/oracle/admin</i>
ORATAB_LOC =<location of the host's oratab file>;	<i>/etc/oratab</i>
PWDFILE_PWD =<common remote sysdba password>	<i>manager</i> <i>(Should be the same on the Primary and Standby Hosts)</i>
PRIMARY_HOSTNAME =<hostname of primary database server>	<i>primary</i>

Appendix A

	<i>(Always the value of host_name in v\$instance)</i>
STANDBY_HOSTNAME =<hostname of standby database server>	<i>standby</i> <i>(Always the value of host name in v\$instance)</i>