|  |  |  |
| --- | --- | --- |
| **MAA - Creating a RAC Physical Standby for a RAC Primary (Doc ID 380449.1)** | [[To Bottom](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=163488668193056&parent=SrDetailText&sourceId=3-12979693561&id=380449.1&_afrWindowMode=0&_adf.ctrl-state=wjv1646mx_21)To Bottom](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=163488668193056&parent=SrDetailText&sourceId=3-12979693561&id=380449.1&_afrWindowMode=0&_adf.ctrl-state=wjv1646mx_21) | https://support.oracle.com/epmos/adf/images/t.gif |

https://support.oracle.com/epmos/adf/images/t.gif

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| \*\*\*Checked for relevance on 11-APR-2014\*\*\*  \*\*\*Checked for relevance on 8-Jul-2015\*\*\*  MAA / Data Guard 10*g* Setup Guide –  Creating a RAC Physical Standby for a RAC Primary  *Oracle Maximum Availability Architecture White Paper*  *May 2006*   **Overview**  [Oracle Maximum Availability Architecture (MAA)](http://www.oracle.com/technetwork/database/features/availability/maa-096107.html) [1] is Oracle's best practices blueprint based on proven Oracle high-availability technologies and recommendations. The goal of MAA is to remove the complexity in designing the optimal high-availability architecture.  Published as part of the MAA series of white papers, this paper focuses on creating a RAC physical standby database for a RAC primary database. This document assumes that there is an existing RAC database and you want to implement Data Guard by adding a standby database to the configuration. The end configuration for this document is a RAC primary database with a RAC standby database. The steps outlined in this document use SQL\*Plus, apply to both Oracle Database 10*g* Release 1 and Oracle Database 10*g* Release 2, and they assume using ASM/OMF, and that the software and ASM instance on the standby host have already been installed/created.  The example used in this document has the database unique name of the RAC database as CHICAGO. The instance names of the two RAC instances are CHICAGO1 (on node chicago\_host1) and CHICAGO2 (on node chicago\_host2). The database unique name of the RAC standby database is BOSTON, and the two standby instance names are BOSTON1 (on node boston\_host1) and BOSTON2 (on node boston\_host2).  This document includes the following tasks:   * Task 1: Gather Files and Perform Back Up * Task 2: Configure Oracle Net on the Standby * Task 3: Create the Standby Instances and Database * Task 4: Configure the Primary Database for Data Guard * Task 5: Verify Data Guard Configuration   This document assumes that the following conditions are met:   * The primary RAC database is in archivelog mode * The primary RAC database is using ASM. * The standby RAC cluster already has ASM instances created * The primary and standby databases are using a flash recovery area. * The standby RAC hosts have existing Oracle software installation. * Oracle Managed Files (OMF) is used for all storage.   **Task 1: Gather Files and Perform Back Up**   1. On the primary node, create a staging directory. For example:   [oracle@chicago\_host1 oracle]$ mkdir -p /opt/oracle/stage   1. Create the same exact path on one of the standby hosts:   [oracle@boston\_host1 oracle]$ mkdir -p /opt/oracle/stage   1. On the primary node, connect to the primary database and create a PFILE from the SPFILE in the staging directory. For example:   SQL> CREATE PFILE='/opt/oracle/stage/initCHICAGO.ora' FROM SPFILE;   1. On the primary node, perform an RMAN backup of the primary database that places the backup pieces into the staging directory. For example:   [oracle@chicago\_host1 stage]$ rman target /  RMAN> backup device type disk format '/opt/oracle/stage/%U' database plus archivelog; RMAN> backup device type disk format '/opt/oracle/stage/%U' current controlfile for standby;   1. Place a copy of the listener.ora, tnsnames.ora, and sqlnet.ora files into the staging directory. For example:   [oracle@chicago\_host1 oracle]$ cp $ORACLE\_HOME/network/admin/\*.ora /opt/oracle/stage   1. Copy the contents of the staging directory on the RAC primary node to the standby node on which the staging directory was created on in step 2. For example:   [oracle@chicago\_host1 oracle]$ scp /opt/oracle/stage/\* oracle@boston\_host1:/opt/oracle/stage  **Task 2: Configure Oracle Net Services on the Standby**   1. Copy the listener.ora, tnsnames.ora, and sqlnet.ora files from the staging directory on the standby host to the $ORACLE\_HOME/network/admin directory on all standby hosts. 2. Modify the listener.ora file each standby host to contain the VIP address of that host. 3. Modify the tnsnames.ora file on each node, including the primary RAC nodes and standby RAC nodes, to contain all primary and standby net service names. You should also modify the Oracle Net aliases that are used for the local\_listener and remote\_listener parameters to point to the listener on each standby host.  In this example, each tnsnames.ora file should contain all of the net service names in the following table:   **Example Entries in the tnsnames.ora Files**   |  |  | | --- | --- | | **Primary Net Service Names** | **Standby Net Service Name** | | CHICAGO = (DESCRIPTION =   (ADDRESS =    (PROTOCOL = TCP)    (HOST = chicago\_host1vip)    (HOST = chicago\_host2vip)    (PORT = 1521))   (CONNECT\_DATA =     (SERVER = DEDICATED)     (SERVICE\_NAME = CHICAGO)      ) ) | BOSTON = (DESCRIPTION = (ADDRESS =    (PROTOCOL = TCP)    (HOST = boston\_host1vip)    (HOST = boston\_host2vip)    (PORT = 1521))   (CONNECT\_DATA =    (SERVER = DEDICATED)    (SERVICE\_NAME = BOSTON)     ) ) |  1. Start the standby listeners on all standby hosts.   **Task 3: Create the Standby Instances and Database**   1. To enable secure transmission of redo data, make sure the primary and standby databases use a password file, and make sure the password for the SYS user is identical on every system. For example:   $ cd $ORACLE\_HOME/dbs $ orapwd file=orapwBOSTON password=oracle  The naming and location of the password file varies on different platforms. See [“Creating and Maintaining a Password File”](http://otn.oracle.com/pls/db102/db102.to_xlink?xlink=ADMIN10241) in the [*Oracle Database Administrator’s Guide*](http://otn.oracle.com/pls/db102/db102.to_toc?partno=b14231) for more information.   1. Copy and rename the primary database PFILE from the staging area on all standby hosts to the $ORACLE\_HOME/dbs directory on all standby hosts. For example:   [oracle@boston\_host1 stage]$ cp initCHICAGO1.ora $ORACLE\_HOME/dbs/initBOSTON1.ora   1. Modify the standby initialization parameter file copied from the primary node to include Data Guard parameters as illustrated in the following table:   **Initialization Parameter Modifications**   |  |  |  | | --- | --- | --- | | **Parameter**  **Category** | ***Before*** | ***After*** | | **RAC Parameters** | \*.cluster\_database=true \*.db\_unique\_name=CHICAGO CHICAGO1.instance\_name=CHICAGO1 CHICAGO2.instance\_name=CHICAGO2 CHICAGO1.instance\_number=1 CHICAGO2.instance\_number=2 CHICAGO1.thread=1 CHICAGO2.thread=2 CHICAGO1.undo\_tablespace=UNDOTBS1 CHICAGO2.undo\_tablespace=UNDOTBS2 \*.remote\_listener=LISTENERS\_CHICAGO CHICAGO1.LOCAL\_LISTENER=LISTENER\_CHICAGO\_HOST1 CHICAGO2.LOCAL\_LISTENER=LISTENER\_CHICAGO\_HOST2 | \*.cluster\_database=true \*.db\_unique\_name=BOSTON BOSTON1.instance\_name=BOSTON1 BOSTON2.instance\_name=BOSTON2 BOSTON1.instance\_number=1 BOSTON2.instance\_number=2 BOSTON1.thread=1 BOSTON2.thread=2 BOSTON1.undo\_tablespace=UNDOTBS1 BOSTON2.undo\_tablespace=UNDOTBS2 \*.remote\_listener=LISTENERS\_BOSTON BOSTON1.LOCAL\_LISTENER=LISTENER\_BOSTON\_HOST1 BOSTON2.LOCAL\_LISTENER=LISTENER\_BOSTON\_HOST2 | | **Data Guard Parameters** |  | \*.log\_archive\_config='dg\_config=    (BOSTON,CHICAGO)' \*.log\_archive\_dest\_2='service=CHICAGO              valid\_for=(online\_logfiles,primary\_role)     db\_unique\_name=CHICAGO' \*.db\_file\_name\_convert='+DATA/CHICAGO/',     '+DATA/BOSTON/','+RECOVERY/CHICAGO',     '+RECOVERY/BOSTON' \*.log\_file\_name\_convert='+DATA/CHICAGO/',     '+DATA/BOSTON/','+RECOVERY/CHICAGO',     '+RECOVERY/BOSTON' \*.standby\_file\_management=auto \*.fal\_server='CHICAGO' \*.fal\_client='BOSTON' \*.service\_names='BOSTON' | | **Other parameters** | \*.background\_dump\_dest=   /opt/oracle/admin/CHICAGO/bdump \*.core\_dump\_dest=     /opt/oracle/admin/CHICAGO/cdump \*.user\_dump\_dest=     /opt/oracle/admin/CHICAGO/udump \*.audit\_file\_dest=     /opt/oracle/admin/CHICAGO/adump \*.db\_recovery\_file\_dest=’+RECOVERY’ \*.log\_archive\_dest\_1 =     'LOCATION=+DATA/CHICAGO/' \*.dispatchers=CHICAGOXDB | \*.background\_dump\_dest=    /opt/oracle/admin/BOSTON/bdump \*.core\_dump\_dest=     /opt/oracle/admin/BOSTON/cdump \*.user\_dump\_dest=     /opt/oracle/admin/BOSTON/udump \*.audit\_file\_dest=     /opt/oracle/admin/BOSTON/adump \*.db\_recovery\_file\_dest=’+RECOVERY’ \*.log\_archive\_dest\_1=     'LOCATION=USE\_DB\_RECOVERY\_FILE\_DEST' \*.dispatchers=BOSTONXDB |    In the above example the primary and standby datafiles are in a single ASM diskgroup. If the primary and standby datafiles are distributed across multiple ASM diskgroups then the unset the DB\_CREATE\_FILE\_DEST parameter prior to starting the standby instance. For further information refer to MetaLink note 340848.1.  For more information about these initialization parameters, see Chapter 13, “Initialization Parameters” in [*Oracle Data Guard Concepts and Administration*](http://otn.oracle.com/pls/db102/db102.to_toc?partno=b14239) manual.  If you are using an SPFILE instead of an initialization parameter file, then see the [“Managing Initialization Parameters Using a Server Parameter File”](http://otn.oracle.com/pls/db102/db102.to_xlink?xlink=ADMIN00202) section in the [*Oracle Database Administrator’s Guide*](http://otn.oracle.com/pls/db102/db102.to_toc?partno=b14231) for instructions on managing an SPFILE.   1. Connect to the ASM instance on one standby host, and create a directory within the DATA disk group that has the same name as the DB\_UNIQUE\_NAME of the standby database. For example:   SQL> alter diskgroup data add directory '+DATA/BOSTON';   1. Connect to the standby database on one standby host, with the standby in the IDLE state, and create an SPFILE in the standby DATA disk group:   SQL> CREATE SPFILE='+DATA/BOSTON/spfileBOSTON.ora' FROM PFILE='?/dbs/initBOSTON.ora';   1. In the $ORACLE\_HOME/dbs directory on each standby host, create a PFILE that is named init*oracle\_sid*.ora that contains a pointer to the SPFILE. For example:   [oracle@boston\_host1 oracle]$ cd $ORACLE\_HOME/dbs [oracle@boston\_host1 dbs]$ echo "SPFILE='+DATA/BOSTON/spfileBOSTON.ora'"> initBOSTON1.ora   1. Create the dump directories on all standby hosts as referenced in the standby initialization parameter file. For example:   [oracle@boston\_host1 oracle]$ mkdir -p $ORACLE\_BASE/admin/BOSTON/bdump [oracle@boston\_host1 oracle]$ mkdir -p $ORACLE\_BASE/admin/BOSTON/cdump [oracle@boston\_host1 oracle]$ mkdir -p $ORACLE\_BASE/admin/BOSTON/udump [oracle@boston\_host1 oracle]$ mkdir -p $ORACLE\_BASE/admin/BOSTON/adump   1. After setting up the appropriate environment variables on each standby host, such as ORACLE\_SID, ORACLE\_HOME, and PATH, start the standby database instance on the standby host that has the staging directoryalter , without mounting the control file.   SQL> STARTUP NOMOUNT   1. From the standby host where the standby instance was just started, duplicate the primary database as a standby into the ASM disk group. For example:   $ rman target sys/oracle@CHICAGO auxiliary / RMAN> duplicate target database for standby;  From 11g onwards we can use, DUPLICATE FROM..ACTIVE DATABASE an alternate option for backup based DUPLICATE  DUPLICATE TARGET DATABASE FOR STANDBY FROM ACTIVE DATABASE   1. Connect to the standby database, and create the standby redo logs to support the standby role. The standby redo logs must be the same size as the primary database online logs. The recommended number of standby redo logs is:   (maximum # of logfiles +1) \* maximum # of threads  This example uses two online log files for each thread. Thus, the number of standby redo logs should be (2 + 1) \* 2 = 6. That is, one more standby redo log file for each thread.  SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 GROUP 5 SIZE 10M, GROUP 6 SIZE 10M, GROUP 7 SIZE 10M;  SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 GROUP 8 SIZE 10M, GROUP 9 SIZE 10M, GROUP 10 SIZE 10M;  These statements create two standby log members for each group, and each member is 10MB in size. One member is created in the directory specified by the DB\_CREATE\_FILE\_DEST initialization parameter, and the other member is created in the directory specified by DB\_RECOVERY\_FILE\_DEST initialization parameter. Because this example assumes that there are two redo log groups in two threads, the next group is group five.  You can check the number and group numbers of the redo logs by querying the V$LOG view:  SQL> SELECT \* FROM V$LOG;  You can check the results of the previous statements by querying the V$STANDBY\_LOG view:  SQL> SELECT \* FROM V$STANDBY\_LOG;  You can also see the members created by querying the V$LOGFILE view:  SQL> SELECT \* FROM V$LOGFILE;  See the [“Configure a Standby Redo Log](http://otn.oracle.com/pls/db102/db102.to_xlink?xlink=SBYDB00426)” section in [*Oracle Data Guard Concepts and Administration*](http://otn.oracle.com/pls/db102/db102.to_toc?partno=b14239) manual for more information.   1. On only one standby host (and this is your designated Redo Apply instance), start managed recovery and real-time apply on the standby database:   SQL> ALTER DATABASE recover managed standby database using current logfile disconnect;   1. On either node of the standby cluster, register the standby database and the database instances with the Oracle Cluster Registry (OCR) using the Server Control (SRVCTL) utility. For example:   $ srvctl add database -d BOSTON –o /opt/oracle/product/10g\_db\_rac $ srvctl add instance -d BOSTON -i BOSTON1 -n boston\_host1 $ srvctl add instance -d BOSTON -i BOSTON2 -n boston\_host2  The following are descriptions of the options in these commands:  The -d option specifies the database unique name (DB\_UNIQUE\_NAME) of the database.  The -i option specifies the database insance name.  The -n option specifies the node on which the instance is running.  The -o option specifies the Oracle home of the database.  Register the ASM instance with the OCR:  $ srvctl add asm -n boston\_host1 -i +ASM1 –o /opt/oracle/product/10g\_db\_rac –p /opt/oracle/product/10g\_db\_rac/dbs/spfile+ASM1.ora  $ srvctl add asm -n boston\_host2 -i +ASM2 -o /opt/oracle/product/10g\_db\_rac –p /opt/oracle/product/10g\_db\_rac/dbs/spfile+ASM2.ora  The following are descriptions of the options in these commands:  The -i option specifies the ASM instance name. If your ASM instance is named +ASM1, then specify it with the ‘+’ included. In crs\_stat output, the resource name will not have the ‘+’ in the resource name. However, the ‘+’ must be specified when an ASM instance name is specified in SRVCTL commands.  The –n option specifies the node name on which the ASM instance is running.  The -o option specifies the Oracle home for the ASM instance.  The -p option specifies the fully-qualified filename of the SPFILE, if the ASM instance is using an SPFILE. This option is not needed if you are using a PFILE located in $ORACLE\_HOME/dbs directory.  The following commands establish the dependency between the database instance and the ASM instance. Again, the ASM instance name must be specified with a ‘+’ if that is the ASM instance name.  $ srvctl modify instance –d BOSTON –i BOSTON1 –s +ASM1  $ srvctl modify instance –d BOSTON –i BOSTON2 –s +ASM2 $ srvctl enable asm -n boston\_host1 -i +ASM1 $ srvctl enable asm -n boston\_host2 -i +ASM2  The following are descriptions of the options in these commands:  The -d option specifies the database unique name (DB\_UNIQUE\_NAME) of the database.  The -i option specifies the database instance name.  The -s option specifies the ASM instance name.  The following commands start, from the OCR standpoint, all ASM instances defined for the node specified by the –n option. If the ASM instance is already running, it will change the status in crs\_stat output from OFFLINE to ONLINE.  $ srvctl start asm –n boston\_host1  $ srvctl start asm –n boston\_host2  **Task 4: Configure The Primary Database For Data Guard**   1. Configure the primary database initialization parameters to support both the primary and standby roles.   \*.log\_archive\_config='dg\_config=(BOSTON,CHICAGO)' \*.log\_archive\_dest\_2='service=BOSTON     valid\_for=(online\_logfiles,primary\_role)     db\_unique\_name=BOSTON' \*.db\_file\_name\_convert='+DATA/BOSTON/',’+DATA/CHICAGO/', ’+RECOVERY/BOSTON’,’+RECOVERY/CHICAGO’ \*.log\_file\_name\_convert='+DATA/BOSTON/',’+DATA/CHICAGO/', ’+RECOVERY/BOSTON’,’+RECOVERY/CHICAGO’ \*.standby\_file\_management=auto \*.fal\_server='BOSTON' \*.fal\_client='CHICAGO' \*.service\_names=CHICAGO  For more information about these initialization parameters, see Chapter 13, “Initialization Parameters” in  the [*Oracle Data Guard Concepts and Administration*](http://otn.oracle.com/pls/db102/db102.to_toc?partno=b14239) manual.  If you are using an SPFILE instead of an initialization parameter file, then see the [“Managing Initialization Parameters Using a Server Parameter File”](http://otn.oracle.com/pls/db102/db102.to_xlink?xlink=ADMIN00202) section in the [*Oracle Database Administrator’s Guide*](http://otn.oracle.com/pls/db102/db102.to_toc?partno=b14231) for instructions on managing an SPFILE.  Note that all the parameters listed above can be dynamically modified with the exception of the standby role parameters log\_file\_name\_convert and db\_file\_name\_convert.  It is recommended to set the parameters with “scope=spfile” so that they can be put into effect upon the next role change.   1. Create standby redo logs on the primary database to support the standby role. The standby redo logs are the same size as the primary database online logs. The recommended number of standby redo logs is one more than the number of online redo logs for each thread. Because this example has two online redo logs for each thread, three standby redo logs are required for each thread.   SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 GROUP 5 SIZE 10M, GROUP 6 SIZE 10M, GROUP 7 SIZE 10M;  SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 GROUP 8 SIZE 10M, GROUP 9 SIZE 10M, GROUP 10 SIZE 10M;  These statements create two standby log members for each group, and each member is 10MB in size. One member is created in the directory specified by the DB\_CREATE\_FILE\_DEST initialization parameter, and the other member is created in the directory specified by DB\_RECOVERY\_FILE\_DEST initialization parameter. Because this example assumes that there are two redo log groups in two threads, the next group is group five.  You can check the number and group numbers of the redo logs by querying the V$LOG view:  SQL> SELECT \* FROM V$LOG;  You can check the results of the previous statements by querying the V$STANDBY\_LOG view:  SQL> SELECT \* FROM V$STANDBY\_LOG;  You can also see the members created by querying V$LOGFILE view:  SQL> SELECT \* FROM V$LOGFILE;  See the [“Configure a Standby Redo Log](http://otn.oracle.com/pls/db102/db102.to_xlink?xlink=SBYDB00426)” section in [*Oracle Data Guard Concepts and Administration*](http://otn.oracle.com/pls/db102/db102.to_toc?partno=b14239) manual for more information.  **Task 5: Verify Data Guard Configuration**   1. On the standby database, query the V$ARCHIVED\_LOG view to identify existing files in the archived redo log. For example:   SQL> select sequence#, first\_time, next\_time from v$archived\_log order by sequence#;   1. On the primary database, issue the following SQL statement to force a log switch and archive the current online redo log file group:   SQL> alter system archive log current;   1. On the standby database, query the V$ARCHIVED\_LOG view to verify that the redo data was received and archived on the standby database:   SQL> select sequence#, first\_time, next\_time from v$archived\_log order by sequence#;  **References**   1. Oracle Maximum Availability Architecturewebsite on OTN  <http://www.oracle.com/technetwork/database/features/availability/maa-096107.html> |