Oracle® Database

Migrating Non-CDBs to New Hardware with the Same Operating System and Release

18c

E97281-02

September 2018

Oracle Database Migrating Non-CDBs to New Hardware with the Same Operating System and Release, 18c

E97281-02

Primary Authors: Sunil Surabhi, Nirmal Kumar

Contributing Authors: Lance Ashdown, Padmaja Potineni, Rajesh Bhatiya, Prakash Jashnani, Douglas Williams, Mark Bauer

Contributors: Roy Swonger, Byron Motta, Hector Vieyra Farfan, Carol Tagliaferri, Mike Dietrich, Marcus Doeringer, Umesh Aswathnarayana Rao, Rae Burns, Subrahmanyam Kodavaluru, Cindy Lim, Amar Mbaye, Akash Pathak, Thomas Zhang, Zhihai Zhang

Contents

Preface

1 Preparing Servers and Network for Database Duplication

2 Preparing the Primary Instance on the Source Host

3 Preparing the Auxiliary Instance on the Destination Host

4 Duplicating the Primary Oracle Database

5 Refresh and Switchover to the Physical Standby Oracle Database

Index

Preface

This guide provides a compilation of topics from the Oracle Database user assistance documentation that are collected to help you complete a specific use case scenario.

- Use Case Scenario for this Document
- Documentation Accessibility

Use Case Scenario for this Document

Use this scenario document to assist you to migrate Oracle Database to new hardware by duplicating an active database to a remote server using RMAN.

While you duplicate the database on your physical standby Oracle Database instance, you do not need to shut down your primary Oracle Database instance. The database remains fully accessible to users while you are performing duplication.

Prerequisites for this Scenario

- Ensure that the operating system release and the file system are identical on both the source and destination servers.
- Ensure that you have installed the same release Oracle Database software on the destination server, and that it is updated to the same release update and release revision.
- Ensure that the source and duplicate database files use the same directory structure.

Outline for this Scenario

1. **Preparing the Servers and Network for Database duplication**. Use RMAN to prepare the servers and network for duplication.
2. **Preparing the Primary Instance on the Source host**. Use RMAN to prepare your primary database.
3. **Preparing the Auxiliary Instance on the Destination Host**. Complete these procedures to prepare your auxiliary instance for switchover.
4. **Duplicating the Primary Oracle Database Instance**. Use `RMAN DUPLICATE`.
5. **Refresh and switchover to the Physical Standby Oracle Database**. Complete these procedures to switch over to your auxiliary instance.

These steps correspond to the chapters in this document.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

1 Preparing Servers and Network for Database Duplication

Create a standby database by duplicating the active database. RMAN copies the datafiles directly from the primary database to the standby database.

You must mount or open the primary database before running the `RMAN DUPLICATE FROM ACTIVE DATABASE` command.

Prerequisites Specific to Active Database Duplication

When you execute `DUPLICATE` with `FROM ACTIVE DATABASE`, at least one normal target channel and at least one `AUXILIARY` channel are required.

If you do not configure or preallocate channels, RMAN allocates the necessary channels by default. If you configure or manually allocate channels for active duplication with backup sets, ensure that the number of auxiliary channels is greater than or equal to the number of target channels.

When you connect RMAN to the source database as `TARGET`, you must specify a user name and password, even if RMAN uses operating system authentication. The connection to the auxiliary instance must use the same user name and password as the source database connection. The source database must be mounted or open. If the source database is open, then archiving must be enabled. If the source database is not open, then it must have been shut down consistently.

When you connect RMAN to the auxiliary instance, the following rules apply:

- When running RMAN on the same host as the auxiliary instance, you can connect locally without a net service name, provided that you connect using a user name and password, and provided that your `DUPLICATE` command does not include the

PASSWORD FILE clause. The connecting user must have the SYSDBA or SYSBACKUP privilege.

- When connecting remotely, or when using the PASSWORD FILE clause in the DUPLICATE command, you must connect using a net service name. You must first create a password file for the auxiliary instance.

The source database and auxiliary instances must use the same SYS and SYSBACKUP password, which means that both instances must have password files. The password file must contain at least two passwords, for the SYS and SYSBACKUP users. You can start the auxiliary instance and enable the source database to connect to it.

The DUPLICATE behavior for password files varies depending on whether your duplicate database will act as a standby database. If you create a duplicate database that is not a standby database, then RMAN does not copy the password file by default. You can specify the PASSWORD FILE option to indicate that RMAN can overwrite the existing password file on the auxiliary instance. If you create a standby database, then RMAN copies the password file to the standby host by default, overwriting the existing password file. In this case, the PASSWORD FILE clause is not necessary.

You cannot use the UNTIL clause when performing active database duplication. RMAN chooses a time based on when the online data files have been completely copied, so that the data files can be recovered to a consistent point in time.

Checklist for Preparing Active Database Duplication

Ensure that you prepare source and target databases before using RMAN to carry out active database duplication.

Source Oracle Database:

- To migrate the source database, you need the database name, database unique name, listener port, service name, database home patch level, and the password for SYS.
- If you have configured source database with Transparent Data Encryption (TDE), then you need a backup of the wallet and the wallet password to allow database duplication with encrypted data.
- The source database can be either in the open or in the mount state.
  - If the source database is open, then it must be in archivelog mode.

    The source database remains fully accessible to users while you are performing the database duplication. Be prepared to take a slight hit on CPU usage and network bandwidth consumption during datafile duplication.
  - If the source database is in the mount state, then shut it down cleanly before bringing it up to the mount state.

> **Note:**
>
> If you choose to maintain the source database in mount state, then the users cannot access the database.

Target Oracle Database:

- A target database system that supports the same database edition as the source database edition.
- Ensure that you have the target database name, database unique name, auxiliary service name, and applied current database home patch level.
- A free TCP port in the target database to setup the auxiliary instance.

Configuring RMAN Channels for Use in Oracle Database Duplication

RMAN channels perform the primary job of database duplication

Each channel corresponds to an Oracle Database server session that performs the duplication tasks. Depending on the duplication technique, RMAN uses either auxiliary channels or target channels.

Use one of the following methods to configure channels:

- Automatically allocate channels by using the `CONFIGURE` command
- Manually allocate channels by using the `ALLOCATE` command

If no automatic channels are configured, then you can manually allocate at least one channel before you begin the duplication. The `ALLOCATE` command that allocates channels must be in the same `RUN` block as the `DUPLICATE` command.

RMAN can use the same channel configurations on the source database for duplication on the destination host even if the source database channels do not specify the AUXILIARY option.

Configuring Automatic Channels Across File Systems

Configure a set of persistent, automatic channels for use in all RMAN sessions.

This example configures automatic disk channels across two file systems:

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 2;
CONFIGURE CHANNEL 1 DEVICE TYPE DISK FORMAT '/disk1/%U';
CONFIGURE CHANNEL 2 DEVICE TYPE DISK FORMAT '/disk2/%U';
```

Because `PARALLELISM` is set to 2, the following command divides the backup pieces between two file systems:

```
BACKUP DEVICE TYPE DISK
  DATABASE PLUS ARCHIVELOG;
```

The following `LIST` command shows how the data file backup was parallelized:

```
RMAN> LIST BACKUPSET 2031, 2032;

List of Backup Sets
===================

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
2031    Full    401.99M    DISK        00:00:57     19-JAN-07
        BP Key: 2038   Status: AVAILABLE  Compressed: NO  Tag: TAG20070119T100532
        Piece Name: /disk1/24i7ssnc_1_1
```

```
  List of Datafiles in backup set 2031
  File LV Type Ckp SCN    Ckp Time  Name
  ---- -- ---- ---------- --------- ----
  1       Full 973497     19-JAN-07 /disk3/oracle/dbs/t_db1.f
  5       Full 973497     19-JAN-07 /disk3/oracle/dbs/tbs_112.f

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
2032    Full    133.29M    DISK        00:00:57     19-JAN-07
        BP Key: 2039   Status: AVAILABLE  Compressed: NO  Tag: TAG20070119T100532
        Piece Name: /disk2/25i7ssnc_1_1
  List of Datafiles in backup set 2032
  File LV Type Ckp SCN    Ckp Time  Name
  ---- -- ---- ---------- --------- ----
  2       Full 973501     19-JAN-07 /disk3/oracle/dbs/t_ax1.f
  3       Full 973501     19-JAN-07 /disk3/oracle/dbs/t_undo1.f
  4       Full 973501     19-JAN-07 /disk3/oracle/dbs/tbs_111.f
```

Configuring Channels for Active Database Duplication

With active database duplication, you need not change your source database channel configuration or configure auxiliary channels. However, you may want to increase the parallelism setting of the source database disk channels so that RMAN copies files over the network in parallel.

The type of active database duplication technique used determines which channels perform the principal work of duplication. When image copies are used to perform active database duplication, the primary work is performed by the target channels. Configure multiple target channels on the source database to improve the duplication performance. When active database duplication is performed by using backup sets, the principal work of duplication is performed by the auxiliary channels. Therefore, it is recommended that you allocate additional auxiliary channels. The number of auxiliary channels must be greater than or equal to the number of target channels. Using backup sets for active duplication also enables parallelism, which can improve the speed of the duplication process.

2 Preparing the Primary Instance on the Source Host

Prepare the primary Oracle Database instance by configuring net services.

- Configuring SQL*Net to Prepare the Primary Instance on the Source Host
  Add entries for primary and standby databases in the `tnsnames.ora` file, and then save the file.

Configuring SQL*Net to Prepare the Primary Instance on the Source Host

Add entries for primary and standby databases in the `tnsnames.ora` file, and then save the file.

Example 2-1 Adding Net Services

```
[oracle @ ora12c-prm ~] $ cd $ ORACLE_HOME / network / admin
[oracle @ ora12c-prm admin] $ cat tnsnames.ora
 # tnsnames.ora Network Configuration File: /u01/app/oracle/product/12.2.0/dbhome/
network/admin/tnsnames.ora
 # Generated by Oracle configuration tools.
 DUPDB =
 (DESCRIPTION =
 (ADDRESS_LIST =
 (ADDRESS = (PROTOCOL = TCP) (HOST = ora12c-dup) (PORT = 1521))
 )
 (CONNECT_DATA =
 (SERVICE_NAME = dupdb )
 )
 )
 PRMDB =
 (DESCRIPTION =
 (ADDRESS = (PROTOCOL = TCP) (HOST = ora12c-prm.localdomain) (PORT = 1521))
 (CONNECT_DATA =
 (SERVER = DEDICATED)
 (SERVICE_NAME = prmdb )
 )
 )
```

3
Preparing the Auxiliary Instance on the Destination Host

RMAN uses an auxiliary instance to create the duplicate database. You must prepare the auxiliary instance before you begin the duplication.

- Installing the Oracle Database Software on the Destination Host
  When the source and destination host are different, you must install the Oracle Database software on the destination host, so that the auxiliary instance can be created.
- Steps to Create an Initialization Parameter File for the Auxiliary Instance
  The initialization parameter file for the auxiliary instance must contain at least the `DB_NAME` and `DB_DOMAIN` initialization parameters. Additional parameters may be specified, if required. Ensure that the initialization parameter file is on the same host as the RMAN client that performs the duplication.
- Copying the Server Parameter File from the Source Database
  If the source database uses a server parameter file, then including the `SPFILE` option in the `DUPLICATE` command directs RMAN to use the server parameter file from the source database for the auxiliary instance.
- Creating a Password File for the Auxiliary Instance
  Connections to the auxiliary instance can be established by using operating system authentication or password file authentication. For backup-based duplication, you can either create a password file or use operating system authentication to connect to the auxiliary instance. For active database duplication, you must use password file authentication.
- Creating Directories for the Duplicate Database
  On the destination host, you must create the directories that RMAN uses to store the duplicate database files on the destination host.
- Using the Same Names for Database Files in the Source Database and Duplicate Database
  Certain conditions must be met to use the same names for files in the source and duplicate database.
- Establishing Oracle Net Connectivity Between the Source Database and Auxiliary Instance
  You must be able to establish a connection between the source database and auxiliary instance for certain forms of duplication.
- Configuring the Network Between Source and Target Oracle Databases
  In preparation for duplicating your Oracle Database, configure the network between your source and target Oracle Database instances.
- Starting the Auxiliary Instance
  The initialization parameter file that you create is used to start the auxiliary instance.

- Making the Oracle Keystore Available to the Destination Host
  If transparent encryption is configured on the source database, then you must ensure that the Oracle software keystore from the source database is available to the auxiliary instance. Manually copy the keystore from the source database to the destination host.
- Starting RMAN and Connecting to Databases
  You must start the RMAN client and connect to the database instances as required by the chosen duplication technique. The RMAN client can be located on any host so long as it can connect to the necessary databases over the network.

Installing the Oracle Database Software on the Destination Host

When the source and destination host are different, you must install the Oracle Database software on the destination host, so that the auxiliary instance can be created.

> **Note:**
>
> Ensure that you install the same release of Oracle Database software, with the same patch level, on both the source and destination host.

Use one of the following techniques to install the software:

- Perform a normal installation with Oracle Universal Installer (OUI).

  Install an Oracle Database whose release number is the same as that of the source database. Do not create a database; install the software only. Apply any required patches.
- Clone the source Oracle home.

  Use OUI to clone the source Oracle home. This ensures that all patches applied to the source database are present in the duplicate database.

Steps to Create an Initialization Parameter File for the Auxiliary Instance

The initialization parameter file for the auxiliary instance must contain at least the `DB_NAME` and `DB_DOMAIN` initialization parameters. Additional parameters may be specified, if required. Ensure that the initialization parameter file is on the same host as the RMAN client that performs the duplication.

To create the initialization parameter file for the auxiliary instance:

1. Do one of the following:
   - Copy the initialization parameter file from the source host to the destination host, placing it in the operating system-specific default location, and then modify the `DB_NAME` and `DB_DOMAIN` initialization parameters.

If you are duplicating a CDB, ensure that the `ENABLE_PLUGGABLE_DATABASE` parameter is present and set to `TRUE`.

See Copying the Server Parameter File from the Source Database.

- Complete these steps:

  a. Using a text editor, create an empty file for use as a text-based initialization parameter file, and save it in the operating system-specific default location.

  b. In the parameter file, set the `DB_NAME` and `DB_DOMAIN` initialization parameters. These are the only required parameters.

     Setting the `DB_DOMAIN` parameter enables you to connect to the default database service when you connect with a net service name.

  c. If the auxiliary instance is to be a CDB, then set the following parameter:

     ```
     ENABLE_PLUGGABLE_DATABASE=TRUE
     ```

2. Set the various location parameters such as `CONTROL_FILES` and `DB_RECOVERY_FILE_DEST`.

3. If necessary, set other initialization parameters like those needed for Oracle Real Application Clusters.

4. Set the required environment variables, such as `ORACLE_HOME` and `ORACLE_SID`.

5. (Optional) Set initialization parameters that specify the location of the duplicate database files if one of the following conditions is satisfied:

   - The source host and the destination host are the same (duplication to the local host).
   - The duplicate database uses a directory structure that is different from that of the source host to store database files.

   Depending on the technique used to specify alternate names for duplicate database files, include one or more of the following parameters in the initialization parameter file: `CONTROL_FILES`, `DB_FILE_NAME_CONVERT`, `LOG_FILE_NAME_CONVERT`, `DB_CREATE_FILE_DEST`, `DB_CREATE_ONLINE_FILE_DEST_`*n*, and `RECOVERY_FILE_DEST`.

   > **Note:**
   >
   > It is recommended that you verify that all paths specified are accessible to the destination host and to the server session of the auxiliary instance.

   See *"Methods of Generating Database File Names for the Duplicate Database"* in *Oracle Database Backup and Recovery User's Guide*.

6. Start SQL*Plus and connect to the auxiliary instance as a user with `SYSDBA` or `SYSBACKUP` privileges. Start the auxiliary instance in `NOMOUNT` mode. If the file is in the default location, no `PFILE` parameter is required on the `STARTUP` command.

   ```
   SQL> STARTUP NOMOUNT;
   ```

Example 3-1   Sample Initialization Parameter File for the Auxiliary Instance

```
DB_NAME=dupdb
CONTROL_FILES=(/dup/oracle/oradata/prod/control01.ctl,
```

```
                dup/oracle/oradata/prod/control02.ctl)
DB_FILE_NAME_CONVERT=(/oracle/oradata/prod/,/dup/oracle/oradata/prod/)
LOG_FILE_NAME_CONVERT=(/oracle/oradata/prod/redo,/dup/oracle/oradata/prod/redo)
```

Copying the Server Parameter File from the Source Database

If the source database uses a server parameter file, then including the SPFILE option in the DUPLICATE command directs RMAN to use the server parameter file from the source database for the auxiliary instance.

For backup-based duplication, the server parameter file is restored from backups. For active database duplication, the server parameter file is copied from the source database to the auxiliary instance.

When the source database uses a text-based initialization parameter file, use the PFILE clause in the DUPLICATE command to copy the source database's initialization parameter file to the auxiliary instance.

You can modify the values that were copied or restored from the server parameter file of the source database by using the PARAMETER_VALUE_CONVERT option of SPFILE or the SET clause of the DUPLICATE. For example, you can use the SET clause to change the value of the DB_FILE_NAME_CONVERT parameter in the auxiliary instance's server parameter file.

If the source database does not use a server parameter file or RMAN cannot restore a backup of the server parameter file, then you must manually create a text-based initialization parameter file, as described in Steps to Create an Initialization Parameter File for the Auxiliary Instance.

Creating a Password File for the Auxiliary Instance

Connections to the auxiliary instance can be established by using operating system authentication or password file authentication. For backup-based duplication, you can either create a password file or use operating system authentication to connect to the auxiliary instance. For active database duplication, you must use password file authentication.

To connect to a database by using password file authentication, you must create a password file for the database. When duplicating to a remote host, setting up a password file is mandatory. The default location for the password file is *$ORACLE_BASE*\database on Windows and *$ORACLE_BASE*/dbs on Linux and UNIX.

> **Note:**
>
> When you create a standby database by using RMAN duplication, password files are always copied. In all other cases, password files are copied only if you specify the PASSWORD FILE option in the DUPLICATE command.

Use one of the following options to create a password file for the auxiliary instance on the destination host:

- Use operating system-specific utilities to copy the source database password file to the destination host and then rename it to match the auxiliary instance name. This is applicable only if the source and destination hosts are on the same platform.
- Create the password file manually. Ensure that the password for the `SYSDBA` and `SYSBACKUP` users are the same in the source database and auxiliary instance.
- Create the password file with the `orapwd` utility. The `SYSBACKUP` option creates a `SYSBACKUP` entry in the new password file.

  The following example creates a password file in the 12.2 format names `orapworcl` that is located in the default location in an operating system file:

  ```
  orapwd FILE='/u01/oracle/dbs/orapworcl' FORMAT=12.2
  ```
- Specify the `PASSWORD FILE` option on the `DUPLICATE... FROM ACTIVE DATABASE` command.

  RMAN copies the source database password file to the destination host and overwrites any existing password file for the auxiliary instance. This technique is useful if the source database password file has multiple passwords to make available on the duplicate database.

  When you use active database duplication, the password file must contain at least two passwords, for the `SYS` user and the `SYSBACKUP` user. These passwords must match the passwords in the source database.

  > **Note:**
  >
  > If you create a standby database with the `FROM ACTIVE DATABASE` option, then RMAN always copies the password file to the standby host.

> **See Also:**
>
> *Oracle Database Administrator’s Guide*

Creating Directories for the Duplicate Database

On the destination host, you must create the directories that RMAN uses to store the duplicate database files on the destination host.

This includes the directories that store the data files, control files, online redo log files, and temp files.

Use the `NOFILENAMECHECK` clause to indicate that RMAN must not display an error when the names of the database files are the same in the source and duplicate database.

Using the Same Names for Database Files in the Source Database and Duplicate Database

Certain conditions must be met to use the same names for files in the source and duplicate database.

The simplest duplication strategy is to configure the duplicate database to use the same directory structure and file names as the source database. You can use the same directory structure and names only when duplicating to a remote host.

Using the same directory structure and file names means that your environment meets the following requirements:

- If the source database uses ASM disk groups, then the duplicate database must use ASM disk groups with the same names.
- If the source database files are Oracle Managed Files, then the auxiliary instance must set the `DB_CREATE_FILE_DEST` parameter to the same directory location as the source database. Although the directories are the same on the source and destination hosts, Oracle Database chooses the relative names for the duplicate files.
- If the names of the database files in the source database contain a path, then this path name must be the same in the duplicate database.
- For Oracle Real Application Clusters (RAC) environments, use the same value for the `ORACLE_SID` parameter of the source and destination databases.

When you configure your environment as suggested, no additional configuration is required to name the duplicate files.

Establishing Oracle Net Connectivity Between the Source Database and Auxiliary Instance

You must be able to establish a connection between the source database and auxiliary instance for certain forms of duplication.

If any of the following conditions is true, the auxiliary instance must be available through Oracle Net Services:

- The RMAN client is run from a host other than the destination host
- The duplication technique chosen is active database duplication
- The destination host is different from the source host

To perform active database duplication, you must connect to the auxiliary instance with `SYSDBA` or `SYSBACKUP` privilege and by using a net service name. The source database to which RMAN is connected as `TARGET` uses this net service name to connect directly to the auxiliary database instance.

To establish Oracle Net connectivity and set up a static listener:

- Follow the instructions in *Oracle Database Administrator’s Guide* to configure a client for connection to a database and add static service information for the listener.

Example 3-2 Example: Establishing Oracle Net Connectivity Between the Source Database and Auxiliary Instance

Assume that the DB_NAME of the source database is src and the source host is src.example.com. The DB_NAME of the auxiliary instance is dup and the auxiliary instance is created on the host dup.example.com.

Use the following steps to establish Oracle Net connectivity between the source database and the auxiliary instance:

1. In the tnanames.ora file of the source database, add the following entry that corresponds to the duplicate database:

   ```
   dupdb = (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dup.example.com)(PORT=1521))
   (CONNECT_DATA=(SERVICE_NAME=dup)))
   ```

2. On the destination host, create the tnsnames.ora file in the $*ORACLE_HOME*/admin/network folder. Add the following entry that corresponds to the source database.

   ```
   srcdb = (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=src.example.com)(PORT=1521))
   (CONNECT_DATA=(SERVICE_NAME=src)))
   ```

Configuring the Network Between Source and Target Oracle Databases

In preparation for duplicating your Oracle Database, configure the network between your source and target Oracle Database instances.

- Adding Static Service to Listener
  Insert a static entry for the standby database in the listener.ora file, and save the file.
- Configuring SQL*Net to Prepare the Auxiliary Instance on the Destination Host
  Add entries for primary and standby databases in tnsnames.ora, and then save the file.
- Checking SQL*Net Configuration

Adding Static Service to Listener

Insert a static entry for the standby database in the listener.ora file, and save the file.

Example 3-3 Adding Static Service to Listener

```
[oracle @ ora12c-dup admin] $ cat listener.ora
 # listener.ora Network Configuration File:
 /u01/app/oracle/product/12.2.0/dbhome/network/admin/listener.ora
 # Generated by Oracle configuration tools.
  SID_LIST_LISTENER    =
 (SID_LIST =
 (SID_DESC =
 (GLOBAL_DBNAME = dupdb )
 (ORACLE_HOME = /u01/app/oracle/product/12.2.0/dbhome)
 (SID_NAME = dupdb )
 )
 )
 LISTENER =
 (DESCRIPTION_LIST =
 (DESCRIPTION =
```

```
 (ADDRESS = (PROTOCOL = TCP) (HOST = ora12c-dup.localdomain) (PORT = 1521))
 )
 (DESCRIPTION =
 (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1521))
 )
 )
 ADR_BASE_LISTENER = / u01 / app / oracle
[oracle @ oel62-ora12c-dup admin] $ lsnrctl status
```

Configuring SQL*Net to Prepare the Auxiliary Instance on the Destination Host

Add entries for primary and standby databases in `tnsnames.ora`, and then save the file.

Example 3-4 Adding Net Services

```
[oracle @ ora12c-dup ~] $ cd $ ORACLE_HOME / network / admin
[oracle @ ora12c-dup admin] $ cat tnsnames.ora
 # tnsnames.ora Network Configuration File: /u01/app/oracle/product/12.2.0/dbhome/
network/admin/tnsnames.ora
 # Generated by Oracle configuration tools.
 DUPDB =
 (DESCRIPTION =
 (ADDRESS_LIST =
 (ADDRESS = (PROTOCOL = TCP) (HOST = ora12c-dup) (PORT = 1521))
 )
 (CONNECT_DATA =
 (SERVICE_NAME = dupdb )
 )
 )
 PRMDB =
 (DESCRIPTION =
 (ADDRESS = (PROTOCOL = TCP) (HOST = ora12c-prm.localdomain) (PORT = 1521))
 (CONNECT_DATA =
 (SERVER = DEDICATED)
 (SERVICE_NAME = prmdb )
 )
 )
```

Checking SQL*Net Configuration

Run the `tnsping` command from the source and destination servers.

```
[oracle @ ora12c-prm admin] $ tnsping standby-db-unique-name
[oracle @ ora12c-dup admin] $ tnsping primary-db-unique-name
```

Oracle recommends that you connect remotely as `sysdba` to ensure that the communication is established properly.

Start SQL*Plus using a command in the following format:

```
sqlplus {username | /} [as sysdba]
```

For example:

```
$ sqlplus / AS SYSDBA
Enter password: password
```

Alternatively, to start SQL*Plus connected to a database other than the default, enter the SQL*Plus command in the format:

```
$> sqlplus username/password@connect_identifier
```

Starting the Auxiliary Instance

The initialization parameter file that you create is used to start the auxiliary instance.

RMAN shuts down and restarts the auxiliary instance as part of the duplication. Hence, it is a good idea to create a server-side initialization parameter file for the auxiliary instance in the default location. If you do not have a server-side initialization parameter file in the default location, then you must specify the client-side initialization parameter file with the `PFILE` parameter on the `DUPLICATE` command.

> **Note:**
>
> Because the auxiliary instance does not yet have a control file, you can only start the instance in `NOMOUNT` mode. Do not create a control file or try to mount or open the auxiliary instance.

To start the auxiliary instance:

1. Start RMAN.

   ```
   % rman
   ```

2. Connect to the auxiliary instance as a user with the `SYSDBA` or `SYSBACKUP` privilege. The following example uses password file authentication to connect to the auxiliary instance.

   ```
   RMAN> CONNECT SYS@dupdb AS SYSDBA;
   ```

   The following example uses operating system authentication to connect to the auxiliary instance by using the `SYSBACKUP` privilege.

   ```
   RMAN> CONNECT / AS SYSBACKUP;
   ```

3. Start the auxiliary instance in `NOMOUNT` mode.

   ```
   RMAN > STARTUP FORCE NOMOUNT;
   ```

Making the Oracle Keystore Available to the Destination Host

If transparent encryption is configured on the source database, then you must ensure that the Oracle software keystore from the source database is available to the auxiliary instance. Manually copy the keystore from the source database to the destination host.

The Oracle software keystore contains the TDE master key used to:

- decrypt encrypted backups when performing backup-based duplication.
- decrypt database or tablespace data when performing active database duplication of TDE-encrypted databases or tablespaces.

The following are the requirements for the keystore at the duplicate database:

- The keystore must be in the default location, or in the location indicated by the `sqlnet.ora` file.
- Permissions on the Oracle keystore file must be set so that the database can access the file.
- During duplication, the auxiliary instance is restarted thereby causing the Oracle software keystore to become unavailable. To ensure that the auxiliary instance has access to the keystore, set the `ENCRYPTION_WALLET_LOCATION` parameter in the `sqlnet.ora` file such that it points to the keystore location.
- With Oracle Real Application Clusters (Oracle RAC), register the auxiliary instance statically with an Oracle Grid Infrastructure listener and use the `ENVS` parameter in the `sqlnet.ora` file of the Oracle Grid home to specify environment variables that set the keystore location and the unique name of the database.

  The following example sets the `ENVS` parameter in `sqlnet.ora` to specify the keystore location and unique database name:

  ```
   (ENVS="ORACLE_UNQNAME=cdbrptl, ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)
  (METHOD_DATA=(DIRECTORY=/etc/ORACLE/WALLETS/cdbrpt1)))")
  ```

- If the source database uses a password-based software keystore (not an auto-login software keystore), then you must provide the keystore password before you begin the duplication.

  Use the `SET` command with the `DECRYPTION WALLET OPEN IDENTIFIED BY` clause to specify the password that must be used to open the keystore.

  The following command specifies the password used to open the keystore (where *`password`* is a placeholder for the actual password that you enter):

  ```
  SET DECRYPTION WALLET OPEN IDENTIFIED BY password;
  ```

> **See Also:**
>
> - *Oracle Database Advanced Security Guide* for information about specifying the Oracle keystore location in `sqlnet.ora`
> - *Oracle Database Advanced Security Guide* for information about the default Oracle keystore location
> - *Oracle Database Advanced Security Guide* for information about converting a standard Oracle keystore to an auto-login keystore
> - *Oracle Database Backup and Recovery Reference* for information about the `SET` command

Starting RMAN and Connecting to Databases

You must start the RMAN client and connect to the database instances as required by the chosen duplication technique. The RMAN client can be located on any host so long as it can connect to the necessary databases over the network.

To start RMAN and connect to the target and auxiliary instances:

1. Start the RMAN client on any host that can connect to the necessary database instances.

   For example, enter the following command at the operating system prompt on the destination host:

   ```
   % rman
   ```

2. At the RMAN prompt, run CONNECT commands for the database instances that are required for your duplication technique.

   > **Note:**
   >
   > When you duplicate a whole CDB or one or more PDBs, connect to the root of both instances.

   - For active database duplication using image copies, you must connect to the source database as TARGET and to the auxiliary instance as AUXILIARY. You must supply the net service name to connect to the AUXILIARY instance. A recovery catalog connection is optional. On both instances, the password for the user performing the duplication must be the same. Any user with a SYSDBA or SYSBACKUP privilege can perform duplication.
   - For active database duplication using backup sets, you must connect to the source database as TARGET by using a net service name. The auxiliary instance uses this net service name to connect to the source database and retrieve the backup sets that are required for the duplication. Connect to the auxiliary instance as AUXILIARY. If you are connecting to the auxiliary instance remotely or intend to use the PASSWORD FILE option of the DUPLICATE command, then connect to the auxiliary instance with a net service name. On both instances, the password for the user performing the duplication must be the same. Any user with a SYSDBA or SYSBACKUP privilege can perform duplication. A recovery catalog connection is optional.
   - For backup-based duplication *without* a target connection, you must connect to the auxiliary instance as AUXILIARY and the recovery catalog as CATALOG.
   - For backup-based duplication *with* a target connection, you must connect to the source database as TARGET and the auxiliary instance as AUXILIARY. A recovery catalog is optional.
   - For backup-based duplication without target and recovery catalog connections, you must connect to the auxiliary instance as AUXILIARY.

   > **Note:**
   >
   > You cannot connect as TARGET to a standby database.

Example: Connecting to the Required Databases When Performing Active Database Duplication

In this example, a connection is established to the source database and the auxiliary instance using net service names. The Net Service name of the source database is srcdb and that of the auxiliary instance is dupdb.

To connect to required databases from the destination host:

1. Start the RMAN client on the destination host.

   ```
   % rman
   ```

2. Connect to the source database as `TARGET`.

   ```
   RMAN> CONNECT TARGET sys@srcdb;
   ```

   Enter the password for the `SYS` user on the source database when prompted.

3. Connect to the auxiliary instance as `AUXILIARY`.

   ```
   RMAN> CONNECT AUXILIARY sys@dupdb;
   ```

   Enter the password for the `SYS` user on the auxiliary instance when prompted.

4

Duplicating the Primary Oracle Database

Complete the steps to duplicate the entire primary Oracle Database to the auxiliary Oracle Database instance on the destination server.

- Verifying the Environment
  Complete the steps before you begin duplication.
- Running the RMAN DUPLICATE Command
  Run the DUPLICATE command on the destination server.
- Postmigration Verification
  Verify if the database has been restored and recovered on the destination server.

Verifying the Environment

Complete the steps before you begin duplication.

1. Ensure that the source database host is reachable from the destination host. You must be able to SSH between the two hosts.
2. On the destination host, use the TNSPING utility to verify the listener port on the source host works fine.

   For example:

   ```
   tnsping source host:1521
   ```
3. On the destination host, use Easy Connect to verify the connection to the source database:

   ```
   host:port/servicename
   ```

   For example:

   ```
   sqlplus system@ip-address:1521/proddb
   ```

   Ensure that the connection string does not exceed 64 characters.
4. Copy the required `sqlpatch` files (for rollback) from the source database home to the target database.
5. Ensure that at least one archivelog has been created on the source database; otherwise, RMAN duplication fails with an error.
6. If the source database uses wallets, then back up the password-based wallet and copy it to the standard location on the destination host.

   For example:

   ```
   /opt/oracle/dcs/commonstore/wallets/tde/db_unique_name/
   ```
7. Make sure the compatibility parameters in the source database are set to at least 11.2.0.4.0 for an 11.2.0.4 database and at least 12.1.0.2.0 for a 12.1.0.2 database.

Running the RMAN DUPLICATE Command

Run the DUPLICATE command on the destination server.

For example:

```
DUPLICATE DATABASE TO dupdb
FOR STANDBY
FROM ACTIVE DATABASE
PASSWORD FILE
SPFILE PARAMETER_VALUE_CONVERT='/app/dbhome1','/app/db_home2'
SET db_file_name_convert='/app/dbhome1/dbs','/app/db_home2/database/dbs'
SET log_file_name_convert='/app/dbhome1/log','/app/db_home2/logfiles'
NOFILENAMECHECK;
```

Postmigration Verification

Verify if the database has been restored and recovered on the destination server.

For example:

`V$DATABASE` displays information about the database from the control file.

```
[oracle @ ora18c-dup admin] $ sqlplus

SQL> select name, open_mode, dbid, created from v$database;
```

`V$INSTANCE` displays the state of the current instance.

```
SQL> select instance_name, host_name from v$instance;
```

`V$DATAFILE` displays information about the datafiles from the control file.

`V$CONTROLFILE` displays the names of the control files.

`V$LOGFILE` contains information about redo log files.

`V$TEMPFILE` displays temp file information.

```
SQL> select name from v$datafile
union
select name from v$controlfile
union
select member from v$logfile
union
select name from v$tempfile;
```

`V$VERSION` displays the version number of Oracle Database.

```
SQL> select banner from v$version;
```

5

Refresh and Switchover to the Physical Standby Oracle Database

Refresh the physical standby Oracle Database with the changes made to the primary Oracle Database, and then switch roles.

- Steps to Refresh a Physical Standby Database with Changes Made to the Primary Database
  Use the RECOVER STANDBY DATABASE command with the FROM SERVICE clause to refresh a physical standby database with changes that were made to the primary database.
- Performing a Switchover to a Physical Standby Database
  These steps describe how to perform a switchover to a physical standby database.

Steps to Refresh a Physical Standby Database with Changes Made to the Primary Database

Use the RECOVER STANDBY DATABASE command with the FROM SERVICE clause to refresh a physical standby database with changes that were made to the primary database.

This example assumes that the DB_UNIQUE_NAME of the primary database is MAIN and its net service name is primary_db. The DB_UNIQUE_NAME of the standby database is STANDBY and its net service name is standby_db.

To refresh the physical standby database with changes made to the primary database:

1. Ensure that the following prerequisites are met:
   - Oracle Net connectivity is established between the physical standby database and the primary database.

     You can do this by adding an entry corresponding to the primary database in the tnsnames.ora file of the physical standby database.
   - The password files on the primary database and the physical standby database are the same.
   - The COMPATIBLE parameter in the initialization parameter file of the primary database and physical standby database is set to 12.0.
2. Start RMAN and connect as target to the physical standby database. It is recommended that you also connect to a recovery catalog.

   The following commands connect as TARGET to the physical standby database and as CATALOG to the recovery catalog. The connection to the physical standby is established using the sbu user, who has been granted SYSBACKUP privilege. The net service name of the physical standby database is standby_db and that of the recovery catalog is catdb.

```
CONNECT TARGET "sbu@standby_db AS SYSBACKUP";
CONNECT CATALOG rman@catdb;
```

3. Roll forward the physical standby database using the `RECOVER STANDBY DATABASE` command with the `FROM SERVICE` clause.

   The `FROM SERVICE` clause specifies the service name of the primary database using which the physical standby must be rolled forward. The standby database is restarted after the roll forward operation.

   The following example rolls forward the physical standby database using the primary database whose service name is `primary_db`.

   ```
   RECOVER STANDBY DATABASE FROM SERVICE primary_db;
   ```

4. (For Active Data Guard only) Perform the following steps to recover redo data and open the physical standby database in read-only mode:

   ```
   ALTER DATABASE RECOVER MANAGED STANDBY DATABASE UNTIL CONSISTENT;
   ALTER DATABASE OPEN READ ONLY;
   ```

5. Start the managed recovery processes on the physical standby database.

   The following command starts the managed recovery process:

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

   When using Data Guard Broker, use the following command to start the managed recovery process:

   ```
   DGMGRL> edit database standby_db set state='APPLY-ON';
   ```

> **See Also:**
>
> *Oracle Database Net Services Administrator's Guide* for information about establishing Oracle Net connectivity

Performing a Switchover to a Physical Standby Database

These steps describe how to perform a switchover to a physical standby database.

> **Note:**
>
> If there is a far sync instance (or a combination of preferred and alternate far sync instances) connecting the primary and standby databases, then the procedure to switchover to the standby is the same as described in this topic. Whether the far sync instances are available or unavailable does not affect switchover. During switchover, the primary and standby must be able to communicate directly with each other and perform the switchover role transition steps oblivious of the far sync instances. See “Using Far Sync Instances” in *Oracle Data Guard Concepts and Administration* for examples of how to set up such configurations correctly so that the far sync instances can service the new roles of the two databases after switchover.

1. Verify that the target standby database is ready for switchover.

   The new switchover statement has a `VERIFY` option that results in checks being performed of many conditions required for switchover. Some of the items checked are: whether Redo Apply is running on the switchover target; whether the release version of the switchover target is 12.1 or later; whether the switchover target is synchronized; and whether it has MRP running.

   Suppose the primary database has a `DB_UNIQUE_NAME` of `BOSTON` and the switchover target standby database has a `DB_UNIQUE_NAME` of `CHICAGO`. On the primary database `BOSTON`, issue the following SQL statement to verify that the switchover target, `CHICAGO`, is ready for switchover:

   ```
   SQL> ALTER DATABASE SWITCHOVER TO CHICAGO VERIFY;
   ERROR at line 1:
   ORA-16470: Redo Apply is not running on switchover target
   ```

   If this operation had been successful, a `Database Altered` message would have been returned but in this example an `ORA-16470` error was returned. This error means that the switchover target `CHICAGO` is not ready for switchover. Redo Apply must be started before the switchover operation.

   After Redo Apply is started, issue the following statement again:

   ```
   SQL> ALTER DATABASE SWITCHOVER TO CHICAGO VERIFY;
   ERROR at line 1:
   ORA-16475: succeeded with warnings, check alert log for more details
   ```

   The switchover target, `CHICAGO`, is ready for switchover. However, the warnings indicated by the `ORA-16475` error may affect switchover performance. The alert log contains messages similar to the following:

   ```
   SWITCHOVER VERIFY WARNING: switchover target has dirty online redo logfiles that
   require clearing. It takes time to clear online redo logfiles. This may slow
   down switchover process.
   ```

   You can fix the problems or if switchover performance is not important, those warnings can be ignored. After making any fixes you determine are necessary, issue the following SQL statement again:

   ```
   SQL> ALTER DATABASE SWITCHOVER TO CHICAGO VERIFY;
   Database altered.
   ```

   The switchover target, `CHICAGO`, is now ready for switchover.

2. Initiate the switchover on the primary database, `BOSTON`, by issuing the following SQL statement:

   ```
   SQL> ALTER DATABASE SWITCHOVER TO CHICAGO;
   Database altered.
   ```

   If this statement completes without any errors, proceed to Step 3.

   If an error occurs, mount the old primary database (`BOSTON`) and the old standby database (`CHICAGO`). On both databases, query `DATABASE_ROLE` from `V$DATABASE`. There are three possible combinations of database roles for `BOSTON` and `CHICAGO`. The following table describes these combinations and provides the likely cause and a high level remedial action for each situation. For details on specific error situations, see “Troubleshooting Oracle Data Guard” in *Oracle Data Guard Concepts and Administration*.

Value of DATABASE_ROLE column in V$DATABASE	Cause and Remedial Action
`BOSTON` database is primary, `CHICAGO` database is standby	Cause: The `BOSTON` database failed to convert to a standby database role. Action: See the alert log for details on the error that prevented `BOSTON` from switching to a standby role, take the necessary actions to fix the error, reopen one of the nodes of `BOSTON` if necessary, and repeat the switchover process from Step 1.

`BOSTON` database is standby, `CHICAGO` database is standby	Cause: The `CHICAGO` database failed to convert to a primary database role.
	Action: Issue the following SQL statement to convert either `BOSTON` or `CHICAGO` to a primary database:
	`SQL> ALTER DATABASE SWITCHOVER TO` *`target_db_name`*
	`FORCE;`
	For example:
	• On the `CHICAGO` database, issue the following SQL statement to convert it to a primary database:
	`ALTER DATABASE SWITCHOVER TO CHICAGO FORCE;`
	• On the `BOSTON` database, issue the following SQL statement to convert it to a primary database:
	`ALTER DATABASE SWITCHOVER TO BOSTON FORCE;`
	If the SQL statement fails with an `ORA-16473` error, then you must start Redo Apply before reissuing the command.
	Restart Redo Apply as follows:
	`SQL> ALTER DATABASE RECOVER MANAGED STANDBY`
	`DATABASE DISCONNECT;`
	Reissue the switchover command as follows:
	`SQL> ALTER DATABASE SWTICHOVER TO BOSTON FORCE;`
	`Database altered.`

`BOSTON` database is standby, `CHICAGO` database is primary	Cause: The `BOSTON` and `CHICAGO` databases have successfully switched to their new roles, but there was an error communicating the final success status back to `BOSTON`. Action: Continue to Step 3 to finish the switchover operation.

3. Issue the following SQL statement on the new primary database, `CHICAGO`, to open it.

   ```
   SQL> ALTER DATABASE OPEN;
   ```

4. Issue the following SQL statement to mount the new physical standby database, `BOSTON`:

   ```
   SQL> STARTUP MOUNT;
   ```

   Or, if `BOSTON` is an Oracle Active Data Guard physical standby database, then issue the following SQL statement to open it read only:

   ```
   SQL> STARTUP;
   ```

5. Start Redo Apply on the new physical standby database. For example:

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

Index