DELL EMC

DELL EMC APPSYNC INTEGRATION WITH ORACLE DATABASE SERVER

A Detailed Review

ABSTRACT

This white paper offers an in-depth look at how DELL EMC® AppSync® integrates with Oracle Database Server. The paper provides detailed information about how AppSync interacts to create, mount, and restore copies of Oracle data. It also provides details about how AppSync manipulates the database instance and other Oracle pieces, and describes AppSync's integration with Oracle ASM and RAC.

April 2018

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Oracle Database Server is at the heart of some of the largest enterprise applications in the world. This application stores critical data that represents tremendous value to countless organizations. DELL EMC AppSync integrates with selected versions of Oracle Database Server and can provide unprecedented protection for critical data.

For the latest information on the specific versions supported by AppSync, please refer to the DELL EMC AppSync Support Matrix. To access the DELL EMC AppSync Support Matrix, go to http://elabnavigator.EMC.com, select **See more**, under **DATA PROTECTION AND AVAILABILITY SOLUTIONS**, and then select **AppSync Support Matrix**. A direct link is here: https://elabnavigator.emc.com/vault/pdf/EMC_AppSync.pdf.

INTRODUCTION

This white paper reflects functionality up to, and including, AppSync version 3.7. It delves into the internal management aspects of DELL EMC AppSync, with a strong focus on Oracle Database Server integration. AppSync integrates directly with Oracle and the underlying host operating system, during copy creation, mount, recovery, and restore operations.

This paper answers questions about how AppSync interacts with Oracle's unique database environment to provide value-add functionality such as database and log consistent copy automation, database repurposing, backup protection, and data redundancy. A certain basic knowledge of AppSync and Oracle concepts is required to benefit from this detailed white paper.

AUDIENCE

This white paper has been written for Oracle DBAs and other technologists who want an in-depth understanding of how AppSync interacts, and integrates, with Oracle Database Server.

APPSYNC OVERVIEW

This overview provides general information about AppSync components and their roles as a basis for more specific discussions that follow. For a more complete overview of AppSync, refer to the *AppSync User and Administration Guide* found on https://support.emc.com/products/25364_AppSync.

AppSync is composed of three main components:

- **AppSync Server** — Stores all information about users, hosts, storage systems, copies, and ongoing operations.
- **AppSync User Interface** — The user interface that allows customers to interact with and control the product. This component includes a command line interface and REST API support.
- **AppSync Plug-in** — Interact with the application and storage layers to create, mount, restore, or expire copies of mission-critical data.

The AppSync plug-in interacts with the Oracle application and issues commands and queries which ultimately affect Oracle and the storage environment in ways that make it possible for AppSync to create, mount and restore copies. The AppSync plug-in resides on the Oracle production host where the Oracle Database Server is running, and any host that must serve as a mount host.

There are several AppSync components that deal with a specific aspect of the environment. The storage services component and the application plug-in, are the primary focus of this white paper.

KEY APPSYNC OPERATIONS

The AppSync Oracle application plug-in manages the interface between AppSync and the Oracle database. Some of the key operations that AppSync performs includes:

- Discovering Oracle databases and tablespaces - Discovery enables AppSync to present a choice of application objects to copy.
- Subscribing Oracle databases to service plans - Enables policy driven protection of Oracle databases using the underlying array technology.
- Running service plans - AppSync enables scheduled, or on-demand, execution of SLA driven plans which allow for protection, mount, and mount with recovery of copied Oracle databases.
    - Full discovery of Oracle objects - AppSync decomposes tablespaces into datafiles, file systems, or raw volumes, and then actual storage (for example, LUNs).
    - Communication with the Oracle database - prepares for the consistent copy operation and resumes normal production operations.
    - Creation of the copy – uses the native storage array technology to create the copy where the Oracle database resides.
- Mount and recover of copies — AppSync can facilitate mounting of the copy as part of the job, and optionally after the copy has been created and mounted, the copy can be recovered in an automated fashion.
- Restoring copies — AppSync can restore a copy in order to revert back to a specific point in time at which the copy was taken.

The following sections provide detailed information about how AppSync performs these tasks.

THE APPSYNC VALUE PROPOSITION

AppSync allows customers to select which Oracle objects to copy, and how to copy, mount, and recover the copies, by defining the various workflows through either a user interface, or UI, command line interface, or CLI, or through REST API commands. The copy management options differ based upon different types of use cases, or required workflows. The following sections outline three possible scenarios and the value proposition offered by AppSync with each scenario.

OPERATIONAL RECOVERY

At its core, AppSync as a product enables quick and reliable protection of vital application data. Users of AppSync may find that creating consistent copies of databases is useful for proactively preparing for situations where unexpected downtime or disasters occur, and one or more databases must be restored quickly. AppSync offers array level recovery, but also integrates with the application layer, providing the most robust protection solution. Operational recovery means that a series of copies can be chosen, maintained on the array, for which a restore operation can be selected. This solution can offer the lowest RTO (recovery time objective) as many points-in-time can be created, and restored from, rather than restoring a copy taken perhaps 24 hours ago, then rolling logs, elongating the recovery process.

BACKUP ACCELERATION

Many customers use AppSync to create a copy of a production Oracle database, mount that copy to another host, and then run a backup against that mount host to an alternate location; other than the primary storage array. This reduces the load on the production server by offloading the backup processes to a non-production mount host.

To satisfy the requirements of this scenario, the copy created should be **roll-forward capable** so the production database can be restored later to any point in time using the archived Oracle logs; applied after the restore.

Oracle backup technology provides a **hot backup** mode, which ensures the consistency of the data, even while the database remains online. Therefore, to support the non-invasive backup use case, customers can choose to create a copy with Oracle online in hot backup mode.

REPURPOSING

A third typical use case workflow, is to create copies of the production Oracle database, in order to mount and repurpose the copy for activities such as test and development, offline reporting, performance testing, training, to name but a few. To accomplish this, AppSync offers an entire area of the product devoted to the creation and management of repurposed copies. These copies can be multi-generational as well, where a 1st generation or 2nd generation removed from production can be repurposed. This means that AppSync can create a copy directly from production, placing the database in hot backup mode if desired, and then offering the ability to create many 2nd generation copies of that copy, thus providing many copies of the same environment, at the exact same point-in-time. There are many use cases which this helps support, such as creating a gold copy for a number of developers to reference and refresh from, that is not changing under production load. Another use case for multi-generation copies is data masking, where the 1st generation copy is mounted, the sensitive data is masked and then unmounted, and finally creating a 2nd generation copy which is presented to developers. This prevents the developers from having access to sensitive data.

It is not recommended to create or refresh second generation copies if the first generation copy has been mounted and recovered. This is because on recovery, the data in the first generation copy undergoes some change. Since a second generation copy is just a storage level copy, a true crash consistent copy, it does not involve quiescing of the production database. Any changes made to first generation during recovery can impact the consistency and recovery of the second generation copy. This usually leads to recovery failures during mount and recovery of second generation copies. For this reason, it is also imperative that the 1st generation copy is not mounted at the time the 2nd generation copy is created.

CONFIGURING THE ORACLE ENVIRONMENT

It is important to follow certain configuration best practices when setting up an Oracle environment in anticipation for AppSync. Following best design practices ensures smooth operation once AppSync is configured. This section explains those best practices.

ORACLE DATABASE DISCOVERY WITH APPSYNC

AppSync relies on the **/etc/oratab** file for discovering Oracle databases on a UNIX host. The file is present as part of **Oracle Restart** functionality and it contains entries in the format **$ORACLE_SID:$ORACLE_HOME:**<N|Y>. An entry is automatically created if databases are created using the Oracle **dbca** utility, however, databases that are created using scripts must use the **srvctl add database** command to add an entry in the **/etc/oratab** file. The following are best practices for avoiding any discovery challenges with AppSync:

1. A duplicate entry, or an entry in an incorrect format, must be avoided.
2. The **$ORACLE_HOME** value must match the exact home location from where the database has been started. This is very important if a soft link is used as ORACLE_HOME (common in SAP with Oracle environments).
3. In case of Oracle **RAC databases**, Oracle adds the **database name** instead of the **ORACLE_SID** in the **/etc/oratab** file. The user need not add entries for individual RAC instances because AppSync is capable of determining the actual instance on each of the RAC nodes that are registered with it.
4. If a database is deleted on the host, the corresponding entry must be removed from the **/etc/oratab** file as well. AppSync continues to display the database in an **offline state** if the entry in the **/etc/oratab** file persists post deletion of the database.
5. If the database id (**DBID**) of a database is changed on the host, but the database name remains the same, AppSync perceives it as a new database altogether. This leads to displaying two entries in the AppSync UI with the same name. If the user intends to get rid of the old entry, one must remove all copies and subscriptions from the old database, comment out the entry in the **/etc/oratab** file (using a **#**) and run the Oracle **Discover Databases** option under **Copy Management**, from within the AppSync UI. At this point, both the entries will be removed from the UI. Now remove the **#** preceding the entry in the **/etc/oratab** file and re-run a discovery. Only the new database will be visible thereafter.
6. AppSync adds an entry in the **/etc/oratab** file for a mounted and recovered database using the **srvctl add database** command. This operation can succeed for Oracle 11g R2 configurations only when **Oracle Grid** is installed on the mount host.
7. If an entry corresponding to a database is deleted from the **/etc/oratab** file, AppSync treats that database as deleted and either marks that database as **pending delete**, if copies and subscriptions are present, or removes the entry altogether from UI.

UNDERSTANDING THE ORACLE LAYOUT AND IDENTIFYING OBJECTS FOR COPY

When copying a file system that is located on a logical volume, AppSync copies, mounts, and restores by volume group. Copies based on volume groups have the following characteristics:

- All of the devices in the volume group are copied
- All of the devices in the volume group are imported
- On restore, all of the devices in the volume group are restored
- This concept applies not just to volume groups, but to ASM diskgroups and VNX, Unity, XtremIO, and RP consistency groups. All disks that are part of that group will be protected and restored if applicable
- When utilizing the **hot backup** feature (default option) archive logs, and optionally the FRA must not share the same file system, volume group, RP or array consistency group, datastore, or volume (LUN) as the data files, control files and/or redo logs.
- If the control files, redo logs, and archive logs and/or FRA share the same as above, then deselect the hot backup method – this method does not protect the archive logs/FRA.

For example, if there are two file systems on two different devices, and both devices reside in a single volume group, it is impossible to restore just one of the file systems, as all files systems belonging to the same volume group are restored. The process begins with all of the logical volumes being deported from the production data server; then after the restore, all of the volumes are imported to the production data.

In this example, the data files, meaning the control files and redo logs, are part of one volume group (VG1), and the archive logs, and optionally the Fast Recovery Area (FRA) files, make up another volume group (VG2). If choosing to copy only the datafiles, the redo logs are automatically copied as they are part of the same volume group. Similarly, if copying the archive logs, for example when using the **hot backup** option, the archive logs, and FRA, or other files in VG2, are managed together.

NOTE: IF OTHER FILE SYSTEMS THAT ARE NOT PART OF THE ORACLE DATABASE ENVIRONMENT, HAPPEN TO SHARE THE SAVE VOLUME GROUP AS THE DATAFILES AND/OR ARCHIVE LOGS, THEN THOSE WILL BECOME EFFECTED ENTITIES AND CAN CAUSE MONTING AND RESTORE ISSUES, AS APPSYNC WILL NOT KNOW ABOUT THOSE OTHER FILE SYSTEMS OR ENVIRONMENTS. HOWEVER, AS OF APPSYNC VERSION 3.7, THESE EFFECTED ENTITIES WILL BE DETECTED AND A WARNING WILL APPEAR IN THE UI.

If you choose to restore only the datafiles, the redo logs will also be restored as they are part of the copy. Similarly when you restore the archive logs, all items in that VG must be restored together.

AppSync 3.7 provides an option to select archive log destination(s) for copies created using hot backup. Earlier AppSync versions used to protect all the archive log destinations. This could be seen as a storage overhead because all destinations contain copies of the same archive log files. Therefore, this option is useful when the user wishes to reduce the storage footprint for an Oracle copy.

MOUNTING AN ORACLE COPY OVERVIEW

AppSync can perform mounts of Oracle databases to an alternate mount host, or in certain cases, also mount to an alternate location on the production host. This section describes considerations regarding these mounts.

Most often, the copies AppSync creates are mounted to an alternate host, also known as a mount host, to perform some processing on the database without impacting the production database, or to perhaps serve as a mount point for backup operations to an alternate location. The processing can range from backing up the copy to tape, running queries for reporting, sanity testing, or other activities facilitated by an offline copy of the database which will not affect the performance of the production database.

The mounting options to service any of these particular needs are varied and can be extensive. The following sections detail some of the possible mount scenarios.

KEY STEPS IN MOUNTING A COPY

The AppSync plug-in on the mount host performs the following:

- Performs storage specific operations against the copy, enabling it to be visible and ready to use on the mount host
    - AppSync uses dynamic LUN masking, meaning that the host does not have a requirement to see the target devices ahead of time, except in special cases such as RecoverPoint.

- Imports any volume groups / mounts any file systems and/or ASM diskgroups
- Optionally performs recovery on the Oracle database and opens it
- Optionally runs any post mount scripts

SPECIFICS REGARDING A PRODUCTION HOST MOUNT

A special case of the mount functionality is a production mount. This capability was added in the AppSync product to allow users to mount copies back to a production host, and to an alternate location on that particular host. Usually this would be used to save the overhead of having an extra mount host, or in certain cases, the ability to recover a particular file which was damaged on the production database, rather than having to perform a restore operation of the entire database. This is known as a surgical restore. In such cases, it is very easy to mount a copy to an alternate path on the same production host, selecting certain files to copy manually over the damaged production data (restoring at the file level). Otherwise, AppSync restores at the LUN/VG/CG level.

Because the production host runs the production database, there are restrictions and exceptions associated with this type of mount which limits the available mount options configurable within the AppSync UI.

ORACLE OBJECTS IMPORTED DURING A MOUNT

The following objects are imported to the mount host during a mount operation:

- **Datafiles** — the datafiles making up the tablespaces of the database being mounted are included in the copy operation in addition to their respective file system(s), device(s), volume(s), or ASM diskgroup(s) metadata. The datafiles includes one or more corresponding UNDO tablespaces for each respective instance, however, tempfiles and temporary tablespaces are not copied or backed up.
- **Control file(s)** — the devices containing the current control files are included in the copy operation in addition to their respective file system(s), device(s), volume(s), or ASM diskgroup(s) metadata. These objects will be imported at the time of mount to be used for recovery & backup operations.
- **Online redo logs** — the devices containing the online redo logs are included in the copy operation in addition to their respective file system(s), device(s), volume(s), or ASM diskgroup(s) metadata. These objects will be imported at the time of mount to be used for recovery & backup operations. If hot backup is used, these files will be overwritten following recovery. If crash-consistent backup is used, they will be used to facilitate recovery and then overwritten following a resetlogs operation.
- **Archive logs** — If hot-backup mode is used (which is the default option), the devices containing the archive logs are included in the copy operation in addition to their respective file system(s), device(s), volume(s), or ASM diskgroup(s) metadata.
- **Init file parameters** — AppSync captures all initialization parameters associated with the database instance and copies them over during mount in the form of an init file (This is done regardless of whether the production instance is using a spfile or init file).
- **Fast Recovery Area** – If the Fast Recovery Area is included in the copy, AppSync imports it during the mount operation. If you are including the FRA in the copy in order to use the flashback feature on the mounted copy of the database and also maintain “roll backward” capabilities, then the best option is to use hot backup mode with a mount read write or read only without database rename.

NOTE: FLASHBACK COMMANDS ARE NOT INTEGRATED INTO APPSYNC AND MUST BE PERFORMED MANUALLY ON THE DATABASE. IN ORDER TO PERFORM A FLASHBACK COMMAND, SHUTDOWN THE INSTANCE, BRING THE DATABASE TO A MOUNTED STATE (STARTUP MOUNT) AND RUN THE FLASHBACK COMMAND MANUALLY. REFER TO ORACLE DOCUMENTATION FOR MORE DETAILS.

MOUNT OPERATION

The mount operation is how the copy is to be processed on the mount host. This can be a copy that is simply mounted and left alone, cataloged with RMAN, repurposed, or presented with scripts for manual recovery scenarios. In many cases, all that is required of AppSync is to mount the copy to an alternate location/host, and once that is complete, a third-party backup utility can apply its own processing to complete a backup to an alternate location, however, there are other mounting scenarios require AppSync to perform an automated application recovery of the Oracle database. In other words, presenting a copied database environment, live and ready for

processing, at an alternate location. Different scenarios require this procedure to occur in different ways and AppSync can accommodate different types of scenarios and needs.

The AppSync Oracle plug-in takes into account many parameters when performing these different mounting operations, depicted in the following sections.

Mount on Standalone Server

This is the simplest form of mounting, performing no database recovery, and does not generate any scripts. It is suitable for manual mount operations where the intended use of the mounted copy falls outside of AppSync's primary backup or recovery use cases – such as granular restore via RMAN, or most often, mounting a copy so another application can copy it off primary storage. This option is not supported if the production database resides on ASM disk groups, as there are no file systems to mount. In the case of ASM, one must utilize one of the following options.

Mount on Standalone Server and create RMAN Catalog Entry

RMAN Cataloging Prerequisites

- RMAN catalog database must exist and must have network connectivity to the mount host.
- The **tnsnames.ora** file on the mount host must contain a **tnsalias** that points to the RMAN catalog database where AppSync should catalog the copy.
- The catalog and catalog owner must be created prior to mounting a copy to be cataloged.
    - Production database must be registered in the RMAN catalog before mounting the copy.
- The Oracle version running the RMAN catalog database must be equal to or greater than the highest Oracle version of all production databases registered to that catalog.

Choosing the option to catalog with RMAN

When choosing to catalog with RMAN, AppSync performs the following operations:

- Starts the Oracle instance on the mount host and brings the database to a mounted state
- Database remains unopened
- AppSync automatically catalogs the components of the mounted Oracle copy in the RMAN recovery catalog using the following RMAN commands:

```
catalog datafilecopy
catalog archivelog
```

- AppSync automatically un-catalogs the same components of the Oracle copy from the RMAN recovery catalog when the copy is unmounted.
- Components that are cataloged include:
    - Datafiles
    - Archive logs

This cataloging allows administrators to utilize the following RMAN capabilities in conjunction with the mounted copy:

- View the contents of the cataloged copy(s) using RMAN commands such as:

```
list datafilecopy all;
```

- RMAN cataloging of the copy's contents allows the following operations to be possible from the RMAN command line utility:

  - Perform RMAN individual file restore from the mounted copy. If a datafile has been lost or damaged, it can be restored using the following RMAN command:

    ```
    restore/recover datafile X
    ```

  - Perform RMAN individual tablespace restore from the mounted copy. If a tablespace has been lost or damaged, it can be restored using the following RMAN command:

    ```
    restore/recover tablespace X
    ```

  - Perform RMAN block-level recovery from the mounted copy. For example, a corrupt block can be recovered using the following RMAN command:

    ```
    blockrecover datafile X block Y
    ```

Consult the Oracle RMAN documentation for further help on commands and syntax. RMAN restore operations are possible for as long as the AppSync copy stays mounted/accessible to the production host.

Skip Data Files

AppSync provides the ability to skip cataloging data files. This setting is under the **RMAN Settings** section, after setting the RMAN specific connection settings. This is not enabled by default, meaning, AppSync will catalog the data files.

Using the BCT File with RMAN Incremental Backups

The **Block Change Tracking** (BCT) file can improve incremental backup performance by avoiding complete scans of the data blocks of the source of the backup. In this case, the source of the backup would be the AppSync copy which just got mounted to a host with the Create RMAN Catalog Entry option.

While AppSync does not integrate directly with RMAN backups, rather it facilitate its integration by cataloging the backup. Along with bringing the database to a mounted state and adjusting the datafile paths before cataloging with RMAN, AppSync will also copy the BCT file to the mount host and adjust its location by using the **alter database rename** file SQLplus command.

Whether RMAN decides to use the BCT file to accelerate the incremental backup is transparent to the user.

The following examples below show a simple versions of RMAN backup commands which would, in this case, leverage the BCT file, since it was selected to be included during the copy.

RMAN backup command against the first copy (Traditionally, this would be a level 0 (full) backup):

```
RMAN> run  { backup incremental level=0 database format '/backup/ora55-20100901-001.bkf' tag ‘full’ ;}
```

RMAN backup command against the second copy. This could be a level 1 incremental backup. This backup will leverage the BCT file for increased backup speed

```
RMAN> run  { backup incremental level=1 database format '/backup/ora55-20100901-002.bkf' tag  'inc1';}
```

To verify, query the **v$block_change_tracking** view:

```
SQL> select filename from v$block_change_tracking;

FILENAME

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

/tmp/ORA55/rmbct.dbf

SQL> select count(*) from v$backup_datafile where used_change_tracking='YES';
```

```
COUNT(*)
----------
23
```

Unmounting a Copy Cataloged with RMAN

Whenever AppSync unmounts an Oracle copy mounted using this option, the copy is uncatalogued from RMAN to indicate that it is no longer available for the recovery operation.

Notes and Restrictions

When integrating with RMAN the following restrictions apply to the Catalog with RMAN mount option:

- Copies created without hot backup mode are not eligible for integration with RMAN.
- Copies mounted with RMAN integration cannot be renamed using AppSync's database rename option.
- Read-only copies cannot be cataloged using RMAN.
- AppSync can be configured to skip the cataloging of Oracle data files. This can be accomplished by specifying this option in the mount dialog
- Add backup control file must be selected during create copy

Mount as Standalone and Recover Database

The most common mount operation in AppSync is to mount and recover the database on a mount host. AppSync either applies archive logs (if hot backup mode was used), or recovers the database using the online redo logs (if hot backup mode was not used). AppSync presents a mounted database matching the point in time when the copy was created.

There are a number of options covered in the ***Recovery Settings*** section, which will allow a user of AppSync to customize the behavior of this operation.

Mount on Standalone Server and Prepare Scripts for Manual Database Recovery

The manual recovery option is very similar to the file system mount option in that it does not start the Oracle instance and does not recover the database. It will, however, prepare some steps for the user to manually recover the database if desired.

- Performs the initialization file modifications necessary to adjust to the various path changes affecting components of the Oracle database
- If ASM is involved, AppSync will mount and rename any ASM diskgroups.
- Generates scripts, as follows (please refer to the AppSync User and Administration Guide for more details):
  - **Step-1_DatabaseRename.sql**
  - **Step-1_DatabaseFileRename.sql**
  - **Step-2_RecoverDatabase.rman**
  - **Step-3_RecoverDatabase.sql**
  - **Step-4_OpenDatabase.sql**

This option is used when the database does not need to be recovered/started and the user wants to preserve the original unaltered state of the copy but also have some expanded recovery options available if needed.

MOUNT SETTINGS

This section details selected mount options that are related to Oracle, and provides guidance to help users decide how to configure the copy mount. Some options are dependent upon the intended use case. Settings such as which host to choose, options when RecoverPoint or VPLEX are part of the environment, and other aspects that are not specifically related to Oracle, or have an effect on Oracle, are not depicted. Please refer to the *AppSync User and Administration Guide* for more details on the various settings.

Choosing the appropriate mount operation and mount path, however, are critical as Oracle must have a proper location for mounting, especially when recovering.

Mount to Path

There are three types of mount path options which are applicable for file systems only, not for ASM managed environments. Depending on the host mounting to, some may not apply.

- **Default Path** - points to a root path **/appsync-mounts**, for UNIX hosts, which serves as a mount point, and the original mount point will be appended to this location. If this root path is not acceptable, simply replace the words **Default Path** with the root mount point of preference. This default option can apply when the copy is mounted back to the same production host, alternate node of a cluster, or to an alternate location.

- **Mapped Path** - provides the ability to customize specific file system targets to change, rather than simply changing the root mount point. For instance, changing **/prd/db1** and **/prd/db2** to **/test/db1** and **/test/db2**.

- **Same as original path** - is only available if mounting to an alternate mount host, and uses the same mount points and paths as the original source system.

The alternate path options mounts a file system to a different location from where it was originally located on the production host. This can be achieved changing the mount point's default path, or using the mapped path in the mount phase dialog or mount wizard. This option only applies to file systems and will have no effect on ASM diskgroups.

These options are used to resolve path conflicts when several copies containing the same file system are mounted on the same host. Additionally, when the production host is the mount host, the alternate path option is critical, to avoid collisions with the original production data. For this reason, the alternate path feature is mandatory if the mount host is the production host.

Recovery Settings

There are different settings which only apply when recovering the database, or when creating scripts that will be used manually to recover the database. In addition, there are differences whether choosing read-only, or read-write, as seen in **Figure 1 - Recovery Settings – Read-0nly** and **Figure 2 - Recovery Settings - Read-write**. None of these options apply when mounting as a file system (no recovery), and if they apply when creating an RMAN catalog entry, it will be depicted. Settings are depicted as to which type of mount operation they apply - RMAN cataloging, recovery, preparing scripts for manual recovery, and/or RAC cluster recovery.

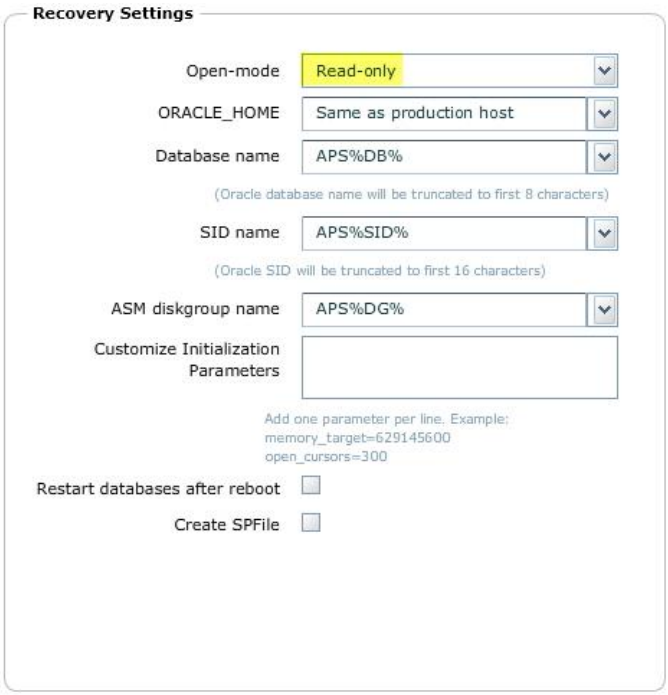

Figure 1 - Recovery Settings – Read-0nly

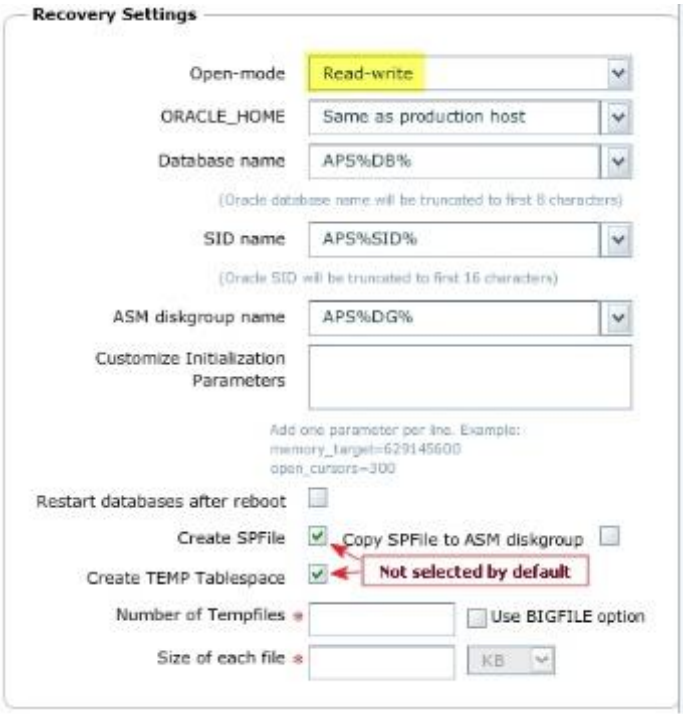

Figure 2 - Recovery Settings - Read-write

- **Open-mode** – Only applies when recovering the database copy, there are two types of modes which Oracle can be recovered, depicted below:
    - **Read-only** - Opens the database in read-only mode, thus not allowing any changes to the copy, other than what is required for the recovery to take place.
        - This mode is used when the user wants to examine data on the mount host, but not alter it in any way.
        - The copy itself is altered, in that it has been recovered and logs have been applied.
        - The actual data contained in the database is not changed.
    - **Read-write** - This mode is almost identical to the **read-only** mode because AppSync performs the same steps to bring up the instance, and update the paths to the relevant components.
        - The major difference relates to how AppSync opens the database at the end of the mount operation. As the name suggests, AppSync opens the database in **read-write** mode, thus allowing further changes to the database once it is mounted.
        - In general, copies that have been mounted and recovered should not be restored to the production database, whether the reset logs clause was used or not. Whenever a database is mounted, there is a possibility that the data may have changed in undesired ways during the mount. This mode is mostly used in repurposing situations where restore is not part of the scenario, such as when working in a testing and development environment.
- **ORACLE_HOME** - Applies when creating an RMAN catalog and all recovery types. Specify the ORACLE_HOME path parameter is strictly used when there is no reliable way for AppSync to validate the accuracy of the selected ORACLE_HOME directly, to be used by the mount host, as it differs from the production host in some way. A user should verify the value for this parameter before making any changes.

  AppSync relies heavily on the mount host's **ORACLE_HOME** to:
    - Find the SQL*Plus utility and run recovery scripts
    - Connect to the Oracle instance being started on the host
    - Check the Oracle version
    - Copy the initialization file in the dbs subdirectory
    - Access any Oracle utilities located in the bin directory of the given ORACLE_HOME path
    - Failure to provide an accurate ORACLE_HOME will result in a range of errors that may not always be easy to diagnose.
- **Database Name** - Only applies when recovering the database copy. The database rename feature is a common option, typically used in repurposing use cases which allows the database to be renamed to something else. This functionality allows multiple copies of the same database to co-exist on a single host and under a single ORACLE_HOME. The %DB% token in the mount options dialog represents the original database name and the user may insert a prefix or postfix (or both) in addition to simply replacing inserting a brand new name not based on the original.
- **SID Name** - Only applies when recovering the database copy. The SID rename feature is rarely used by itself, as it is usually associated with database rename. SID rename changes the name of the Oracle instance (process) that is started on the mount host. SID rename is accomplished by renaming the initialization file and some key parameters inside of it, such as instance_name.

  Since an Oracle SID manages an Oracle database, users often choose to change both for consistency. But the SID rename operation in itself does not alter the copy in any way.

  NOTE: APPSYNC CREATES DIRECTORIES BASED ON THE SID NAME WHEN PERFORMING MOUNTS. THE USE OF THE SID RENAME CAN SOMETIMES BE LEVERAGED TO ALLOW SEVERAL COPIES MOUNTED ON A HOST TO AVOID DIRECTORY STRUCTURE COLLISIONS, BUT WITH NO PLAN TO ACTUALLY START THE DATABASE. IN SUCH CASES, THE USE OF SID RENAME IS NECESSARY

TO AVOID CLASHES OF DIRECTORY STRUCTURES THAT HAVE THE SID NAME EMBEDDED, OR OVERWRITING FILES CONTAINING THE SID NAME, SUCH AS THE INITIALIZATION FILE OR PASSWORD FILE.

One example in which SID rename is useful involves mounts to the production host. For example, a database called **PROD**, with a corresponding SID called **PROD**, can have a copy mounted onto the same production host, yet to an alternate path, without actually starting it. Simply select the **Mount on standalone server and prepare scripts for manual database recovery** option and choose to rename the SID. Rename the SID to **PROD2**, for example, will inform AppSync that the database copy must not be started, and the alternate SID name, **PROD2**, is provided so the directory structure does not collide, or do not clash with, **PROD**. To actually start **PROD2** on the production host, rename the database as well, to prevent a database clash with **PROD** at startup time.

- **ASM Diskgroup Name** - Applies when creating an RMAN catalog and all recovery types. AppSync offers the ability to rename ASM diskgroups using the original name as a token - %DG%.

  In the mount settings, you can apply a prefix or postfix to this value to alter the name of the diskgroups as part of the mount. For example, diskgroups named DG1, DG2, and DG3, can renamed to **NEWDG1**, **NEWDG2**, and **NEWDG3** respectively, by changing this parameter to **NEW%DG%**.

- **Customize Initialization Parameters** - Applies when creating an RMAN catalog and all recovery types. Use the AppSync Console to enter customized initialization parameters for mounting Oracle data. When the mount operation runs, these parameters will be appended to the copy of the initialization parameters used with the mounted database.

  AppSync internally overrides certain init parameters to apply the database changes that are required to mount and/or recover the database copy with the options specified by the user. The parameters specified in customize initialization parameters window should be used to customize any parameters that need to have a different value than production. It is recommended that users refrain from overriding init parameters that AppSync itself overrides, such as the following parameters:

  - log_archive_dest_<n>
  - dbname
  - db_create_file_dest
  - db_create_online_log_dest_<n>
  - control_files
  - local_listener
  - remote_listener
  - db_recovery_file_dest

- **Restart database after reboot** - Applies only when AppSync recovers the database. This check box allows AppSync to add an entry for auto starting Oracle after the mount host is rebooted.

- **Create SPFile** – Only applies to when AppSync recovers the database in a standalone environment, not applicable with RAC. This option creates the SPFile on the mount host. If selected, optionally copy the SPFile to the ASM diskgroup, if applicable.

- **Advanced Recovery Options –** Applicable for standalone recovery only, additional recovery options include creating up to three additional control files, using ADR, DBID, and/or disabling archive log mode.

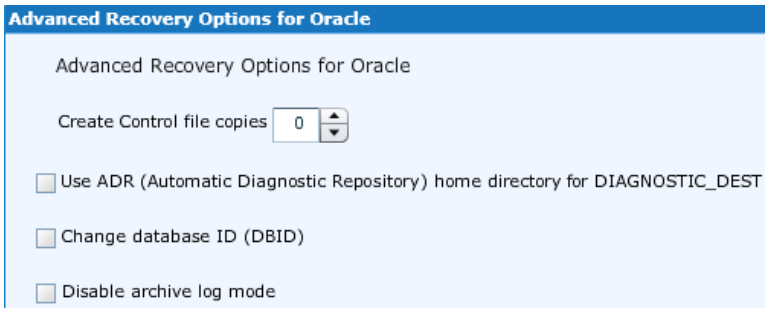

Figure 3 - Advanced Recovery Options

FOR ALL RECOVERY TYPES OF OPTIONS, PLEASE REFER TO THE *APPSYNC USER AND ADMINISTRATION GUIDE* FOR MORE DETAILS.

UNMOUNTING ORACLE COPIES

Unmounting an Oracle copy is the opposite of mounting a copy. When you perform unmount, AppSync completes the following actions:

- Shuts down the running database if it had been started by AppSync. (If the “Mount and generate Scripts for Manual recovery” or “Mount Filesystem” modes were used, AppSync does not attempt to shut down the database; that is the user's responsibility.)
- Dismounts the mounted ASM diskgroups, if ASM is part of the environment. Refer to the section *ASM model with AppSync* for details.
- Unmounts file systems, if file systems are part of the environment.
- Deports/exports third-party volume groups.
- Updates the AppSync server with the necessary information required for AppSync to be able to perform a restore or another mount later on. This is especially critical with information regarding paths, such as datafile paths. If a datafile was mounted using an alternate path, the control file was updated with the new path. Any subsequent mount operations must consider previous path changes, so as to maintain an accurate chain of events.

An ASM instance user with **SYSASM** privileges is required to perform tasks in AppSync. The ASM instance user information provided is used during the mounting phase to perform the **asm_diskstring** and **asm_diskgroup** operations during mount and unmount operations. In cases where **SYSASM** privileges are revoked from the ASM instance user before the unmount operation, the operation fails.

RETRY RECOVERY

Starting AppSync 3.7, users can retry recovery of a mounted a recovered copy without having to unmount the copy. This saves a lot of time that is usually spent in removing the target devices from the host as well as ESX rescans (if VMWare hosts are involved). This feature can be used when:

1. The user wishes to retry recovery with a different set of options like changing the open-mode, specifying a different database name/SID/ASM diskgroup, or perhaps adding customized initialization parameters.
2. Recovery can also be retried if it fails to recover the copies during the initial mount operation. This reduces the amount of time it takes to perform unmount and then re-mount operations. This comes handy, if there is a recovery failure due to a user or host issue because recovery can be retried after rectifying the problem.
3. The recovery mode remains fixed across retries. For example, if the copy was first mounted and recovered as RAC database, then retry recovery cannot be attempted with recover as standalone database option.
4. This option is also available for repurposed copies.

ASM MODEL WITH APPSYNC

AppSync’s Oracle plug-in interacts with the ASM instance in order to discover, mount, and dismount ASM diskgroups. On the production system, during a copy or restore operation, AppSync interacts with the ASM instance serving the database.

For the copy operation, AppSync will check whether an ASM rebalancing operation is in progress for any of the diskgroups involved, and will fail protection if an ongoing operation does not terminate within an hour. This check is mandatory to ensure copy consistency by guaranteeing that there are no data movements going on between the disks of the diskgroups that are being protected by AppSync. AppSync itself does not have control over an ongoing, or user initiated, rebalancing operation, however, it sets the **asm_power_limit** parameter to **0** to prevent any new operation from triggering dynamically, while protection is in progress. The original value of the parameter is restored once protection is completed.

For the mount and recovery operations, AppSync uses UDEV rules based nomenclature in the case of LINUX, and **mknod** in case of IBM AIX, for making target devices available to the mount host’s ASM instance. AppSync adds **/dev/emc-appsync-*** to the **asm_diskstring** parameter of the mount host ASM instance for recognizing the devices mounted by AppSync.

Key points on setting **asm_diskstring** parameter on LINUX mount hosts includes setting the parameter correctly in order to ensure that the ASM instance sees only one path to the target devices. Duplicate paths lead to diskgroup mount failures. The key points are as follows:

1. The **/dev/emc-appsync-*** paths are **UDEV** rules based **NAME** parameter, in the case of RHEL 6.x, or **UDEV** rules based **SYMLINK+ parameter**, in the case of RHEL 7.x.
2. Avoid an empty **asm_diskstring** parameter. This can be validated using **asmcmd dsget** command run as **grid user**.
3. Avoid nested **diskstrings** like **/dev/*** and **/dev/asm-disks***.
4. Avoid generic **diskstrings** like **/dev/***, **/dev/mapper/*** and **/dev/emcpower/***. Generic **diskstrings** will always lead to conflicting paths because the target device will be presented via the generic path as well as via **/dev/emc-appsync-*** path.
5. The standard convention for **ASMLIB** devices is **ORCL:***, however, AppSync can work with **/dev/oracleasm/disk/*** also. Only one of the above conventions can be used for an ASM instance.
6. The standard convention for **ASMFD** devices is **AFD:***.

The steps performed during mount are summarized as follows:

1. Connect to the ASM instance on the mount host using **SQLPlus** and as a user with **SYSASM** privileges using the command:

   ```
   connect / as SYSASM;
   ```

2. Fetch the path of ASM disks on the mount host using the query:

   ```
   select path from v$asm_disk;
   ```

3. Set the **asm_diskstring** parameter to the ASM disks retrieved in the query above. In the example below, it is assumed that **Path1** and **Path2** were the **diskstrings** present on the mount host before the current copy was mounted and the **ORCL:*** disks are those of the copy being mounted.

   ```
   alter system set asm_diskstring = 'Path1','Path2',
   '/dev/emc-appsync-*' scope=both;
   ```

4. Fetch the ASM diskgroups on the mount host using the following query:

   ```
   select name from v$asm_diskgroup;
   ```

5. Set the **asm_diskgroups** parameter to the ASM diskgroups retrieved in the query above. In the example below, it is assumed that **DiskGroup1** and **DiskGroup2** were the diskgroups present on the mount host before the current copy was mounted and **DG1**and **DG2** are the diskgroups of the copy.

   ```
   alter system set asm_diskgroups = DiskGroup1,DiskGroup2,DG1,DG2 scope=both;
   ```

6. Mount the **diskgroups** in the copy using the following command:

   ```
   alter diskgroup DG1 mount;
   alter diskgroup DG2 mount;
   ```

   AppSync automatically performs these steps during mount in **Read-only** and **Read-write** operations. For the mount option **Mount on standalone server and prepare scripts for manual database recovery**, the steps above must be performed by the user manually, if the database should be recovered on the mount host.

Similarly, during unmount, the following steps performed are:

1. Connect to the ASM instance on the mount host using **SQLPlus** and as a user with **SYSASM** privilege using the command:

   ```
   connect asm_username/password as SYSASM;
   ```

2. Fetch the ASM diskgroups on the mount host using the query:

   ```
   select name from v$asm_diskgroup;
   ```

3. Dismount the diskgroups of the mounted copy using the command:

   ```
   alter diskgroup DG1 dismount;
   alter diskgroup DG2 dismount;
   ```

4. Set the **asm_diskgroups** parameter to the ASM diskgroups excluding those of the mounted copy. In the example below, it is assumed that **DiskGroup1** and **DiskGroup2** were the diskgroups present on the mount host before the current copy was mounted.

```
alter system set asm_diskgroups = DiskGroup1,DiskGroup2 scope=both;
```

5. Fetch the path of the ASM disks on the mount host using the following query:

```
select path from v$asm_disk;
```

6. Set the **asm_diskstring** parameter to the ASM disks excluding those of the mounted copy. In the example below, it is assumed that **Path1** and **Path2** were the **diskstrings** present on the mount host before the current copy was mounted.

```
alter system set asm_diskstring = 'Path1','Path2' scope=both;
```

AppSync automatically performs these steps during the unmount process, if the copy was mounted in **Read-only** or **Read-write** with recovery mode.

For the mount option **Mount on standalone server and prepare scripts for manual database recovery**, the steps above must be performed manually by the user.

ASMLIB VOLUMES IGNORED DURING MOUNT

If an ASM disk contains an ASMLIB volume header, this header will be ignored during mount as the disks mounted by DELL EMC AppSync will be blacklisted against ASMLIB. This does not change the physical structure of the disk in any way, and preserves the ASMLIB volume name should it need to be restored back to production.

NOTE: SOMETIMES FOR MOUNT TO PRODUCTION HOST SCENARIO, PRODUCTION ORACLE ASM DISK GROUPS CANNOT BE MOUNTED AFTER A HOST REBOOT DUE TO CONFLICTING ASMLIB DISKS. THIS CAN HAPPEN IF THE UDEV RULES THAT MASK THE DEVICES OF AN APPSYNC MOUNTED COPY DO NOT GET LOADED AFTER A REBOOT LEADING TO CONFLICT BETWEEN THE PRODUCTION ASMLIB DEVICES AND THE MOUNTED COPY'S DEVICES. IF UDEV RULES ARE NOT LOADED, THEN THE MOUNTED COPY'S DEVICES ARE EXPOSED THROUGH THEIR ASMLIB HEADER BECAUSE THAT INFORMATION IS PRESENT ON THE REPLICATED DEVICE AND IT IS NOT HIDDEN BY THE UDEV RULES. THEREFORE, THE ASM INSTANCE SEES TWO ASMLIB DISKS WITH THE SAME NAME AND GETS CONFUSED. THIS CAN BE ADDRESSED EITHER BY UNMOUNTING AND REMOUNTING THE COPY IN APPSYNC OR BY MANUALLY RELOADING THE UDEV RULES (FOLLOW STANDARD COMMANDS AS PER LINUX VERSION AND FLAVOR.).

SUPPORT FOR MOUNTING A RAC ASM COPY TO A TARGET RAC

The following requirements must be satisfied in order to mount a copy as a RAC database:

- The database must reside on ASM diskgroups
- One or more nodes of the target Grid cluster must be added to AppSync. AppSync will only mount to the nodes which are already added.
- AppSync supports mounting to a Grid cluster that is equal to, smaller or larger than the size of the original production cluster.
- If using virtual machines, there is a limitation of about 40 disks which can be used simultaneously to mount a RAC copy. This is due to the first SCSI controller typically being occupied by disks running in no-sharing mode. This leaves only three free controllers which can be used for shared-mode RDMs.

RESTORING AN ORACLE COPY

This section describes points to remember related to the restore of Oracle copies.

RESTORING ORACLE COPIES WITH APPSYNC

AppSync creates copies of Oracle databases and also supports restoring the database back to production. Check the ***What Oracle Objects get Restored*** section below, for details on restoring objects from a copy. The basic restore operations are:

1. Shutdown the database that is to be restored
2. Dismount/deport the underlying ASM DGs/Volume Groups/Filesystems (whatever applicable)
3. Performs the actual array level restore of the data - the production storage is overwritten by the data in the copy
4. Import/remount back the ASM DGs/Volume Groups/Filesystems (whatever applicable)
5. AppSync does not automate the database recovery process, post restore, so it must be done manually by the database administrator

AFFECTED ENTITIES

AppSync's Oracle plug-in analyzes the database and its logical layout on the storage array. For example, if the database is composed of five tablespaces, all of which were included in the copy, AppSync ensures that all of the underlying datafiles, file systems, and volume groups are discovered. When a job runs, AppSync maps the source data to the target devices based on the particular job selection, and copies the data.

IMPORTANT: THE GRANULARITY OF RESTORE OPERATIONS IS TIGHTLY LINKED TO THE UNDERLYING STORAGE LAYOUT ON WHICH THE DATABASE WAS BUILT.

Keep in mind that even two distinct file systems might be associated if they are built on volumes with a common volume group. AppSync copies at the volume group level as portions of a volume group cannot be extracted. Ultimately, the restore granularity matches the granularity dictated by the storage layout on which the database is built.

WHAT ORACLE OBJECTS GET RESTORED

The previous section stressed the importance of storage layout when it comes to restore considerations. This section explains what logical objects get restored and under what circumstances. The options available for restore granularity are:

- Data only - includes data files, control files, and redo logs. The init and/or spfile are not restored. One can use the alert log to rebuild the init file in case it is lost.
- Archive logs only
- Both data and archive logs

ORACLE DATA GUARD SUPPORT

Oracle Data Guard is a disaster recovery solution where a primary database will copy its archived redo logs over a network connection to what is referred to as a standby database. AppSync supports these types of standby databases:

1. Physical Standby
2. Snapshot Standby

These types of standby configurations can be opened in one of two modes:

1. **Active Standby Mode** - When a Standby database is in **read-only** or **read-write** mode
2. **Passive/Non-Active Standby Mode** - When a Standby database is in **mounted** mode

AppSync provides support for Oracle Data Guard configurations for the **primary** (the source database) and the **Physical Standby** (the target database) which is open in **Active** or **Passive/Non-Active Standby** mode. A **Physical Standby** database open in **Active Standby** mode means that the standby database can be opened in **read-only** mode while logs are being applied to it. This allows the database to be queried for information while the logs are being applied. **Snapshot Standby** configurations also allow the database to be open in **read write** mode. A **Passive/Non-Active Standby** mode setting means that the database will be started in **mounted** mode and logs will be applied in the background.

PHYSICAL STANDBY

In a **Physical Standby** environment, the archive logs are applied as they come in. A **Physical Standby** has a 1:1 mapping in terms of the file and storage layout from primary to standby. A typical primary-to-standby (source-to-target) Data Guard configuration can be seen in **Figure 4 - Oracle Data Guard Layout**. A **Physical Standby** database can be open in both **read-only** or **mounted** mode which means it can be either an **Active Standby** or **Passive/Non-Active Standby** configuration.

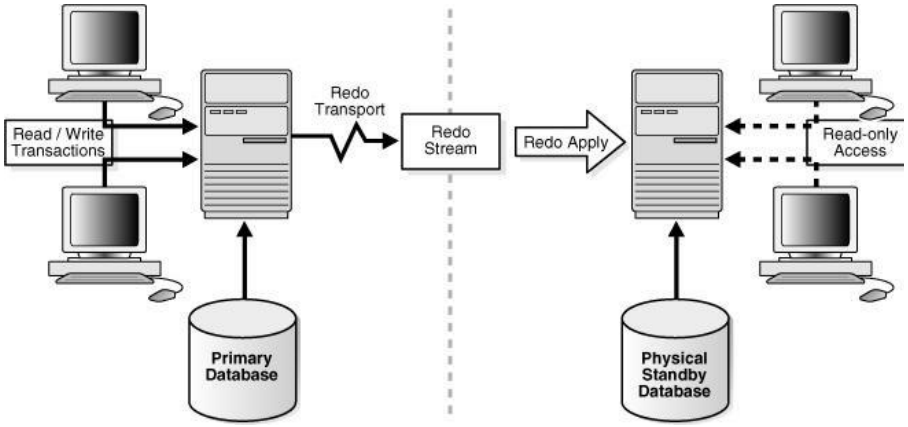

Figure 4 - Oracle Data Guard Layout

SNAPSHOT STANDBY

In a **Snapshot Standby** environment the archive logs are not applied as they come in and are pooled or stored in the archive log directory of the standby until they are all applied. A snapshot standby has a 1:1 mapping in terms of the file and storage layout from primary to standby. A **Snapshot Standby** database is typically open in **read-write** or **Active Standby** mode for data to be accessed. However it has to switched back to **mounted** mode in order for the pooled archive logs to be applied to the database.

DATA GUARD RELATIONSHIP VIEW

Data Guard relationship information is displayed in a column called **Data Guard Relationship**, which will show up if there are databases configured with Data Guard present. The column will display information about the database, including the type of database, the primary database it is associated with, and the corresponding open mode of the database. In **Figure 5 - Data Guard Relationship**, both **PRODDG** and **TIGER** rare **primary** databases, while **PROD_DRDG** and **TIGERSTBY** are the corresponding **Standby** databases. Both are open and in **non-active**, or **mounted** mode.

Figure 5 - Data Guard Relationship

ORACLE DATA GUARD CONSIDERATIONS

There are some considerations while using Data Guard databases with AppSync. These considerations apply during operations with the protection and repurposing workflows,mounting, recovery, and restore.

1. **Protection/Repurposing** – For a database in an Oracle Data Guard configuration, only a **primary** database can be protected using the **hot backup** mode. **Standby** databases cannot be protected using hot backup mode. This is because Oracle does not allow a Standby database to enter hot backup mode. A **Standby** database that is in open in **Passive/Non-Active Standby** mode can be protected **without using hot backup**, without having to change the open mode of the database. This only applies to Data Guard databases however, and not regular Oracle standalone databases.

2. **Mount/Recovery** – Operations like mounting and recovering copies, work the same way for Oracle Data Guard databases, as for regular Oracle Standalone databases. When a Primary or a Standby database is mounted, and then recovered, it loses all its Data Guard properties and behaves like a Standalone Oracle database on the host where it is mounted.

3. **Restore** – For a database in an Oracle Data Guard configuration, the restore operations works the same way as it does for Oracle Standalone databases. Only **Primary** databases, in a Data Guard environment, can be restored using AppSync. **Standby** databases cannot be used for restore purposes.

TURNING A COPY INTO PRODUCTION ORACLE DATABASE

As of AppSync 3.5, a mounted and recovered Oracle database copy can be discovered on a mount host, and then protected as if it were a production Oracle database instance. There are a number of considerations and restrictions, however, so please refer to the following knowledge base article https://support.emc.com/kb/501266. Creating Repurposed copies from such databases is not recommended.

TROUBLESHOOTING ORACLE ISSUES DURING COPY

This section provides information about how to troubleshoot certain issues that customers may encounter when using AppSync with Oracle.

For automatic discovery of Oracle databases, several Oracle queries are run at the start of the copy. Whenever AppSync operations fail with Oracle SQL errors, it is important to get the results of those queries.

DISCOVERY FAILURE OR STATUS DISPLAYED AS OFFLINE

1. Ensure the Oracle database is up and running. Connect to the database as **sysdba** and run the following SQL commands:

```
sqlplus / as sysdba
select * from dba_data_files;
select open_mode from v$database;
```

2. The database must be open **in read-write** mode.
3. Ensure that a valid entry must be present in the **/etc/oratab** file, if the database doesn't appear in AppSync UI.
4. Pluggable databases are not supported, and so are not displayed.

FAILURES DURING COPY CREATION

1. Ensure the database is open and in **read-write** mode.
2. Ensure there are no ongoing **ASM rebalancing** operation on any of the ASM diskgroups that are participating in protection. AppSync will not stop an existing rebalancing operations, but will prevent one from initiating during its copy cycle.
3. Ensure whether the user is trying to protect a container database with pluggable databases attached to it. AppSync as of 3.7 does not support such databases.
4. Ensure no other application is trying to freeze the database at the same time.
5. Ensure hot backup is selected only for databases that have archive logs configured and do not share volumes, volume groups, or consistency groups with control files or redo logs.
6. Ensure all components of the database reside on same DELL EMC primary storage mode. AppSync, as of 3.7, cannot protect federated environments.

CONCLUSION

This white paper explains how to use AppSync with Oracle Database Servers, including how to configure Oracle environments for optimal copy creation. This white paper also describes the details associated with creating a service plan to copy Oracle data with AppSync, such as how to create a copy, including important information about setting credentials, discovering the Oracle environment, planning the copy layout, etc. In addition, this white paper reviews consistency aspects, and advanced Oracle subjects such as copying the Flash Recovery Area (FRA), copying data stored under an Automatic Storage Management (ASM) instance, Data Guard support, restoring Oracle data in varied Oracle configurations, and turning a previous copy into production and making new copies from it.

The white paper describes various mounting techniques in detail, including specifics of how AppSync carries out mounts such as a mount to the production host, the various mounting options affecting the resulting impact on future restores of the mounted copy. Finally, a troubleshooting section provides some solutions to some common issues encountered managing Oracle with AppSync.

REFERENCES

For more information on AppSync, and how to manage Oracle environments, consider the following sources:

DELL EMC AppSync Installation and Configuration Guide

DELL EMC AppSync Security Configuration Guide

DELL EMC AppSync User and Administration Guide

LIST OF FIGURES