White Paper

EMC Replication Manager Integration with Oracle Database Server

A Detailed Review

Abstract

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

September 2011

EMC²

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided “as is.” EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

All other trademarks used herein are the property of their respective owners.

Part Number h4117.5

Table of Contents

Executive summary (Heading 1 paragraph style)

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. EMC® Replication Manager integrates with selected versions of Oracle Database Server and can provide unprecedented protection for this critical data.

For the latest information on the specific models and versions supported by Replication Manager, refer to the EMC Replication Manager Support Matrix. To access the EMC Replication Manager Support Matrix, go to http://elabnavigator.EMC.com/, select **PDFs and Guides**, and scroll down to Replication Manager.

Introduction

This white paper reflects functionality up to Replication Manager 5.4. It delves into the internal aspects of EMC Replication Manager with a strong focus on Replication Manager's deep integration with Oracle Database Server. Replication Manager integrates with Oracle during replica creation, mount, and restore in order to quiesce the database during the replication and capture the necessary logs.

This paper answers questions about how Replication Manager interacts with Oracle's unique database environment to provide value-add functionality such as database repurposing, backup protection, and data redundancy. A certain basic knowledge of Replication Manager and Oracle concepts is required for the reader to benefit from this detailed paper.

Audience

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

EMC Replication Manager overview

This overview provides general information about Replication Manager components and their roles as a basis for more specific discussions that follow. For a more complete overview of Replication Manager, refer to the white paper *EMC Replication Manager Version 5.0 Technology - A Detailed Review* found on EMC.com.

- Replication Manager is composed of three components:
- Replication Manager Server — Stores all information about users, hosts, replicas, and ongoing operations.
- Replication Manager Console — The user interface that allows customers to interact with and control the product. This component includes a command line interface as well.

Replication Manager Agents — Interacts with the application and storage layers to create, mount, restore, or expire replicas of mission-critical data.

The Replication Manager agent interacts with the Oracle application and issues commands and queries that ultimately affect Oracle and the storage environment in ways that make it possible for Replication Manager to create, mount and restore replicas. The Replication Manager agent resides on the Oracle production host where the Oracle Database Server is running.

There are several Replication Manager components that each deal with a specific aspect of the environment. The storage services component and the various application agents are the primary focus of this white paper.

Key Replication Manager operations

The Replication Manager Oracle application agent manages the interface between Replication Manager and your Oracle database. Some of the key operations that Replication Manager performs include:

- Discovering Oracle databases and tablespaces — Discovery enables Replication Manager to present a choice of application objects for replication.
- Creating application sets — Application sets allow users to specify which application objects should be replicated.
- Creating jobs — Jobs define actions to be performed on that application set and that can be run multiple times, using various options and settings. Jobs allow you to perform actions such as:
  - Quiescing data
  - Creating replicas of the application set on certain storage targets
  - Mounting those replicas
  - Running optional customer scripts
- Running jobs — Replication Manager enables scheduled or on-demand execution of previously defined jobs. Tasks that jobs perform include:
  - Full discovery of application objects (tablespaces). Replication Manager decomposes tablespaces into datafiles, file systems, or raw volumes, and then actual storage (for example, LUNs).
  - Communication with the application so that it can prepare for the replication operation and later resume normal functioning.
  - Creation of the replica from the storage where the application resides (actions are dependent on the array technology used).
- Mounting replicas — Replication Manager can facilitate mounts of the replica as part of the job, or after the replica has been created. An optional mount operation can make the replica available on the production or an alternate mount host.

- Restoring replicas — Replication Manager can restore a replica in order to revert back to a specific point in time at which the replica was taken.

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

Application programming interfaces

The Replication Manager Oracle agent uses two application programming interfaces (APIs) to communicate with Oracle, retrieve information it needs to discover where various pieces of the database are stored, and perform certain operations such as "begin backup" and "end backup."

Replication Manager integrates with Oracle to perform the following activities:

- Database queries and internal tests for connections to the database using the Oracle Call Interface (OCI)
- Database operations, such as "alter tablespace begin backup" or "startup mount" or "recover database", using SQL*Plus and RMAN command line utility scripts

The Replication Manager value proposition

Replication Manager allows customers to select which Oracle objects to replicate and define how to perform the replication. The replication options differ based upon the use case. The following sections outline two possible scenarios and the value proposition offered by Replication Manager in these scenarios. Let's explore them in more detail.

Noninvasive backup

Many customers use Replication Manager to create a replica of a production Oracle database, mount that replica to another host, and then run a backup on that mount host. 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 replica created should be "roll-forward capable" so the production database can be restored later to any point in time using the archived Oracle logs.

Replication Manager provides the following options to satisfy this requirement:

- Consistent split online replicas using hot backup mode
- Non-consistent split online replicas using hot backup mode
- Consistent split replicas using offline backup mode
- Non-consistent split replicas using offline backup mode

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 replica with Oracle online in hot backup mode or with Oracle offline during the replication.

Repurposing or cloning with Replication Manager

Customers also create replicas of the production Oracle database to mount and repurpose the replica for activities such as test or offline reporting. This scenario requires a restartable image. To accomplish this, Replication Manager offers Online Consistent Split without Hot Backup mode.

Configuring your Oracle environment

It is important to follow certain configuration best practices when you set up your Oracle environment. Following best practices as you design the Oracle environment ensures smooth operation once Replication Manager is installed and configured. This section explains those best practices and offers links to other required reading.

Finding fundamental configuration information

Before using Replication Manager, verify that Oracle is configured properly for your environment. Fundamental Oracle configuration steps are detailed in the "Oracle Procedures" chapter of the *EMC Replication Manager Product Guide.* Review the section "Configuring Oracle for Replication Manager" for important setup information.

Understanding your layout and identifying objects for replication

When you are replicating a file system that is located on a logical volume, you can replicate, mount, and restore by volume group. Replicas based on volume groups have the following characteristics:

- All of the devices in the volume group are replicated.
- All of the devices in the volume group are imported.
- On restore, all of the devices in the volume group are restored.

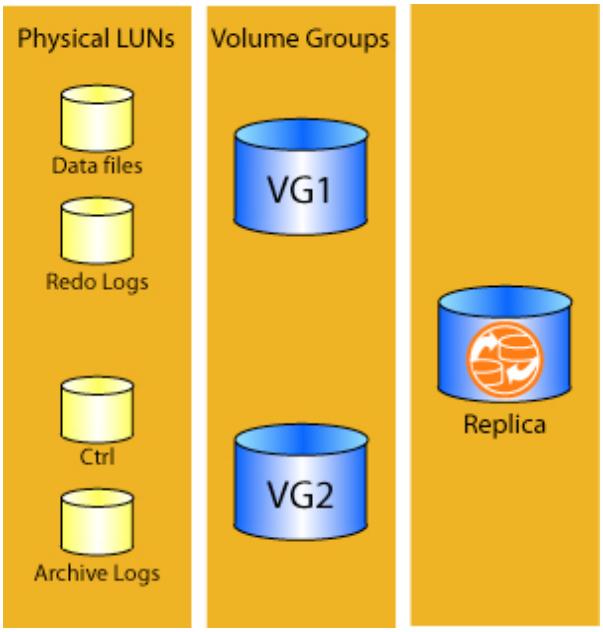

Figure 1. Volume group layout

For example, if there are two file systems on two different devices and both devices reside in a single volume group, if you ask to restore one of the file systems, both file systems will be restored. (First, all of the logical volumes are deported from the

production data server; after the restore, all of the volumes are imported to the production data.)

In this example, the datafiles and redo logs are part of one volume group (VG1), and the control file and archive logs make up another volume group (VG2). If you choose to replicate only datafiles, the redo logs are automatically replicated as they are part of the same volume group as the datafiles. Similarly, if you replicate the control file (for example, when using the Consistent Split option for replication), the archive logs are automatically replicated as they share the same volume group as the control file.

If you choose to restore only the datafiles, the redo logs will also be restored as they are part of the replica. Similarly when you restore the control file, the archive logs are automatically restored overwriting the production archive logs. This is why it is best to configure these pieces on separate volume groups to ensure that restoring one object won't affect the recovery of the data by automatically restoring another object.

Table 1 summarizes the objects that are replicated in the various Replication Options available in Replication Manager. This information can help you determine how to arrange these objects into volume groups.

Table 1. Objects replicated when certain replication options are used

1. Options\Objects	2. Data	3. Redo	4. Ctrl	5. Arch
6. Consistent Split Hot backup	7. ✓	8. ✓	9. ✓	10. Optional
11. Consistent Split No Hot backup	12. ✓	13. ✓	14. ✓	15. Optional
16. Consistent Split (Offline)	17. ✓	18. ✓	19. ✓	20. Optional
21. No Consistent Split Hot backup	22. ✓	23. ✗	24. ✗	25. Optional
26. No Consistent Split Offline	27. ✓	28. ✗	29. ✗	30. Optional
31. No Consistent Split No Hot backup	32. ✓	33. ✗	34. ✗	35. Optional
36. Key: ✓ = Replicated; ✗ = Not replicated				

The following notes also apply:

- For RAC setups (non-ASM) the archive logs should be on an NFS share so that the Replication Manager agent can gather them on the node from which the replication is taking place and send them over to a Replication Manager server. If you choose to replicate the archive log directory (optional), the replication must take place on the specific RAC node where the archive log directory file system is local, as Replication Manager cannot replicate a remotely mounted file system. If the archive log directory is not selected for replication, it can be located anywhere and simply made available to the RAC nodes.
- In Replication Manager versions 5.0.2 and later using ASM-based databases, archived logs may reside in an ASM diskgroup (recommended).
- For hot backup mode replications, the archived logs relevant to the current replica and backup control file are transferred over the network to the Replication Manager server at the end of replication.
- In all cases, a copy of the spfile/initialization file is transferred over the network as well (see later sections).

Creating application sets

Replication Manager application sets of Oracle databases define the part of the Oracle environment to be replicated. In order to do that, Replication Manager must discover the Oracle instance, collect the necessary credentials to gain access to that instance, and collect any information about the Oracle instance necessary to replicate it.

Discovering Oracle instances

The very first contact with the Oracle agent occurs the first time the user selects “Oracle” as the application to be added to an application set. At this point, the agent attempts to discover which Oracle instances (SIDs) are available on the host before it tries to connect to Oracle.

Replication Manager discovers the instance as follows:

- On UNIX-based systems, Replication Manager searches for the oratab file and parses it from top to bottom, extracting the SID name from each line found in the file. Typically a line is formatted as follows: PROD:/oracle home:Y|N
- When parsing the oratab file on UNIX systems, Replication Manager ignores ASM instances, which start with a "+" sign by convention, and only lists database instances since ASM instances in themselves cannot be replicated but only serve as providers of diskgroups to database instances.
- On Windows-based systems, Replication Manager searches for keys in the registry that contain instance names. Each Oracle instance is associated with an NT service. Replication Manager scans the following key to extract service names that match this pattern: OracleService‹sid› under \HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services.

Not all Oracle instances discovered are necessarily running currently. Replication Manager presents the list of instances discovered and the administrator can choose an appropriate instance from the list. With Replication Manager 5.4 onwards, for the RAC-aware support feature, the oracle agent verifies if the instance selected for appset creation is currently running on the node used for appset creation, otherwise an appropriate error is displayed.

Additionally, for UNIX-based clients, there is an Add Instance button (in the Replication Manager Console of Replication Manager version 5.1 and later) that allows the administrator to manually configure a SID not found during initial discovery. For example, if not all SIDs are listed in the oratab file, the administrator can use this button to manually add SIDs.

Database connection and authentication

Once Replication Manager has identified an instance to use, the Replication Manager Console prompts for credentials and other information required to connect to the database instance.

There are three levels of authentication on UNIX-based Oracle configurations and one on Windows-based Oracle configurations as follows:

- **Database user/password (common for UNIX and Windows)** — This user is an RDBMS user created in the Oracle database using the “create user” command. This is the username that Replication Manager uses internally to connect to Oracle, run SQL queries through OCI, and perform other operations with SQL*Plus.

  This user needs to have a minimum of DBA level privileges (needs to be granted the DBA role) for online replications and basic mounts, and SYSDBA privileges to have access to all the functionality of Replication Manager. See *Database user privileges* for more details.

- **Connect String (UNIX and Windows)** — This is the TNS alias that must correspond to one entry in the tnsnames.ora file configured on the host. It will resolve to the server, port, and instance name for which the connection is intended. This alias needs to resolve to a DEDICATED server connection (as opposed to a SHARED connection).

  By default, Replication Manager pre-fills that field with a value that is equal to the name of the instance, as is usually the case. However, the user must ensure that the alias is correct. In Replication Manager version 5.4 and later, in case of RAC databases, Replication Manager pre-fills the field with RAC database name. The oracle agent over-writes this value with RAC instance running on the node.

- **Operating system username (UNIX only)** — This is the operating system username used to install the Oracle binaries on the host. When Replication Manager launches Oracle operations, it creates a sub-process that assumes the ID of that operating system user on the host, so as to provide an appropriate context to run queries and other commands. In releases prior to Replication Manager 5.1, the Oracle operations were run in a process context owned by the root user, which could potentially cause permission issues.

- **TNS_ADMIN (UNIX and Windows)** — This field contains the path to the tnsnames.ora file (and is the equivalent of the Oracle environment variable called TNS_ADMIN), necessary for the translation of the connect string into the host, port, and instance name that Replication Manager should use to connect to the Oracle instance. By default this field is populated with the following: $ORACLE_HOME/network/admin/.

- **ORACLE_HOME (UNIX only)** — This field points to the ORACLE_HOME path. It defaults to the value found in the oratab file for the corresponding SID.

CAUTION: Verifying the TNS alias is even more important in RAC environments. The following paragraph provides an explanation.

Replication Manager version 5.2 and later list the RAC instance name pertaining to the host on which the application set has been defined. For older versions, you can add it using the Add Instance button. If a three-node RAC database is called MYRAC, the instance names would most likely be MYRAC1, MYRAC2, and MYRAC3. When you enter the instance name, verify that you entered the instance that is associated with the RAC node on which you are creating the application set and job. For example, if you select a load-balanced TNS alias to a host on the cluster instead of a host-specific TNS alias pointing to one particular instance of the cluster, your replication may fail because Oracle may try to connect to the wrong node of the cluster at any given time (for example, MYRAC may point to MYRAC1 one moment, and MYRAC2 another time). The problem is that such an alias on RAC setups is usually a load-balanced alias that can potentially connect Replication Manager to any of the RAC nodes. The Replication Manager Oracle agent is not clustered and must work on one specific node at a time, for example, the node where MYRAC1 is running. By giving the connect string “MYRAC”, there is a potential for Replication Manager to connect to MYRAC2, which may cause problems during the replication process, because the job

cannot run on that node. Replication Manager 5.4 and later list the RAC global database name, instead of the RAC instance name pertaining to the host on which the application set has been defined. Once you enter the global database name, the oracle agent determines the RAC instance running on the node using the command:

```
srvctl status database -d <RAC_DB_Name>
```

From the output of the command, the oracle agent searches for the instance running on the host on which the appset is being created.

For Replication Manager versions prior to 5.2.x, follow these steps to safely configure an application set for a RAC database:

1. Click **Add Instance** in the Application Set Wizard.
2. Complete the SID information for the correct active instance name (MYRAC1 from our example)
3. Complete the rest of the Application Set Wizard to create the application set.

In Replication Manager versions 5.2 and later, the appropriate instance that is relevant to the given host where the application set is being created will be listed (MYRAC1 for example). Therefore, it will not be necessary to use the Add Instance button. In Replication Manager versions 5.4 and later, the RAC global database name is listed. In case the RAC database name is not listed, use the Add instance button to add the global database name.

In the case of Oracle 11gR2 RAC configurations, ensure that the tnsnames.ora file has an entry for the instance running on that node. In case an entry is missing, it can be added by following the steps below:

1. Log in as a grid user.
2. From the $ORACLE_HOME/bin directory, stop the listener using the following command:
3. lsnrctl stop
4. Log in as the database user.
5. Go to the $TNS_ADMIN location and edit the tnsnames.ora file and add an entry like the following example:

```
RACDB1 =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST =
SCAN.rmcluster.local)(PORT = 1521))
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = racdb)
(INSTANCE_NAME = racdb1)
)
)
```

Set the parameter INSTANCE_NAME to the Oracle RAC instance running on the node for which to create an application set in Replication Manager. This ensures a connection to the correct instance on the registered node during the creation of the application set. (racdb1 in the example above).

6. Log in as the grid user, go to $ORACLE_HOME/bin directory, and start the listener using the following command:

   ```
   lsnrctl start
   ```

Notes: For 11gR2 RAC configurations, Replication Manager supports the Single Client Access Name (SCAN) feature for use with Oracle RAC; however, the SCAN IP cannot be used to register a host with Replication Manager. Instead, choose the hostname, public IP, or virtual IP.
For Oracle 11g R2 RAC One Node database configuration with Replication Manager 5.4 onwards, please refer section "Support for Oracle 11g R2 RAC One Node Feature"

ASM connection and authentication

The Application Credentials panel also collects information important when connecting to an ASM instance as follows:

- **ASM username/password (UNIX only)** — If the tablespaces that make up the selected database are built on ASM diskgroups (Oracle 10*g* and later), additional credentials are required in order for Replication Manager to connect to the ASM instance and decompose those diskgroups into disks that the storage services layer of Replication Manager can understand and map to the appropriate LUNs.

  Note: The only username for the ASM instance is SYS (Oracle 10*g* through Oracle 11gR1). In Oracle 11gR2, the ASM instance user must be a user with SYSASM privileges. In Oracle 11gR2, the ASM instance user SYS is no longer a requirement; any user with SYSASM privileges is supported for authenticating to the ASM instance.

- **ASM Instance** —The ASM instance name is usually prefilled with “+ASM”, which is the most common instance name. In RAC environments, the ASM instance is usually named +ASM‹*n*› where ‹*n*› is the node number. Replication Manager will generate the appropriate default value corresponding to the RAC node on which the application set is being defined. For example, the default ASM instance name for the second node of the RAC cluster would be +ASM2.

- **ORACLE_HOME** — Replication Manager version 5.2 and later, support the use of two separate ORACLE_HOME directories, one for ASM instances and one for Oracle databases.

  Prior to version 5.2.3, Replication Manager required the same Oracle operating system user and group to access both the database ORACLE_HOME and ASM ORACLE_HOME. Replication Manager version 5.2.3 and later support a separate operating system user and group to access the database ORACLE_HOME and ASM ORACLE_HOME (ORACLE_HOME for Grid installation in case of Oracle 11gR2). The

environment must list the ASM instance separately in the /etc/oratab file.

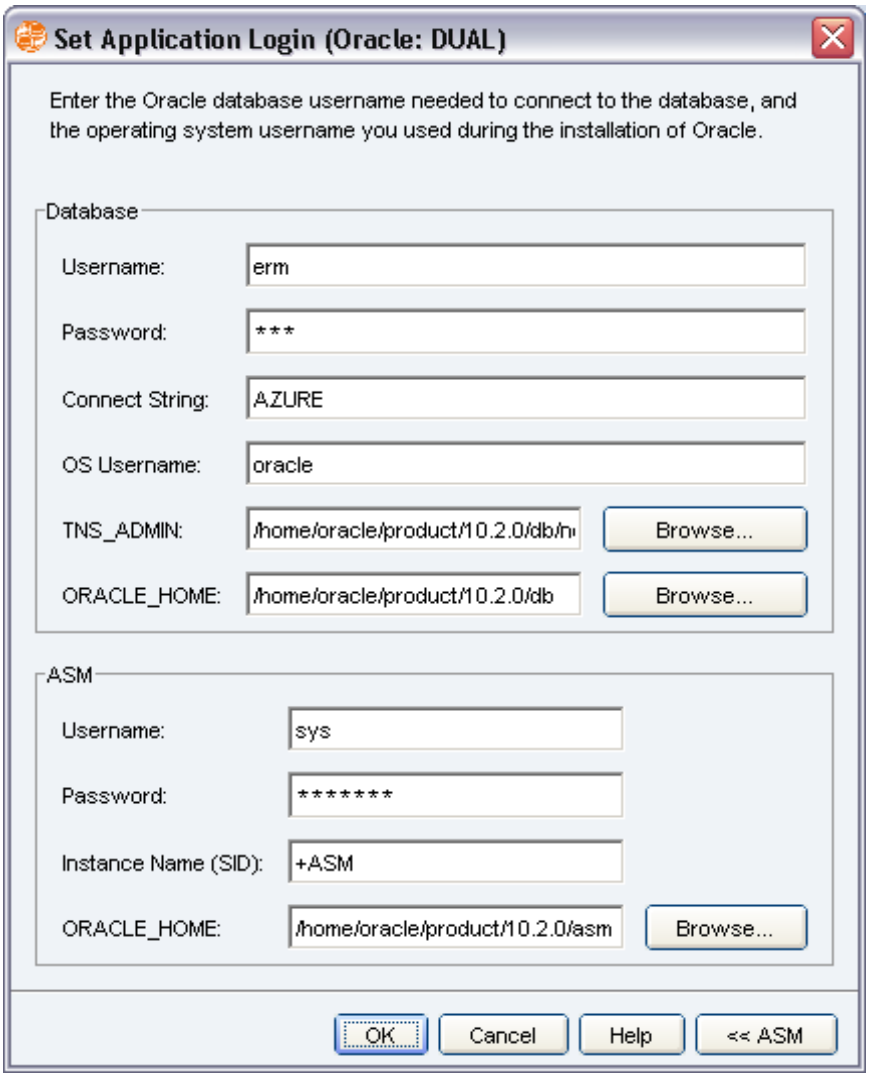

Figure 2. Application set authentication screen

Database user privileges

Replication Manager version 5.1 SP1 and earlier releases require all users of the Oracle agent to possess SYSDBA credentials (the highest level of access for an Oracle database). This facilitates all operations, including online backups, mounting to an alternate host, recovering the database, and restoring a replica over production data. SYSDBA privileges ensure a simple authentication model with adequate privileges to perform all tasks that Replication Manager may attempt.

However, the simplicity of this was not flexible enough in all customer situations. Starting with version
5.1 SP2, Replication Manager allows non-SYSDBA users to participate in the Replication Manager framework in limited ways. A minimum of DBA role is required for Replication Manager to function properly. Access to certain dba_* views is

necessary to facilitate discovery of the Oracle database, and other SQL*Plus commands require many of the privileges included in the Oracle DBA role.

Users without SYSDBA privileges are not permitted to perform the following tasks:

Tasks restricted to SYSDBAs	Reason for restriction
Offline replications	Task requires a shutdown of the database, which cannot be done by non-SYSDBA users.
Selection of read only or read/write recovery options	Task requires starting up the database on the remote host, which cannot be done by non SYSDBA users.
Restoring Oracle replicas	Task requires overwriting of existing data, restricted to SYSDBA users only.
Select the "copy BCT file" replication option	Task requires executing the *dbms_backup_restore.bctswitch()* stored procedure, which cannot be done by a non-SYSDBA user

Note: If a DBA user created a replica, and was later granted SYSDBA privileges, Replication Manager can detect this at the time of restore if the database is available before the restore operation, and allows the restore to proceed.

ASM instance user privileges

For Oracle 10.x and Oracle 11gR1, Replication Manager requires a SYS account in the Username field while creating the application set. For Oracle 11gR2, Replication Manager allows all ASM instance users who possess SYSDBA and SYSASM privileges. ASM instance users without SYSDBA or SYSASM privileges are not permitted to perform any tasks in Replication Manager on ASM databases created in Oracle 11gR2.

Replication Manager uses the SYSDBA connection in OCI calls and the SYSASM clause for sqlplus calls,

Understanding the Oracle process

The Replication Manager client delegates Oracle operations and queries to a child process (oracle_process on UNIX, oracle_process.exe on Windows) spawned at the beginning of each operation (replication, restore, mount, application set creation, and so on). On Windows 2003 systems, the Replication Manager client is always a 32-bit application, which can run on either 32- or 64-bit Windows. The oracle_process.exe, however, is a 64- or 32-bit application depending on which version of Windows is running. This allows Replication Manager to support 32- and 64-bit Oracle on Windows. On a Windows 2008 platform, both the Replication Manager client and oracle_process pieces are always native 64-bit processes.

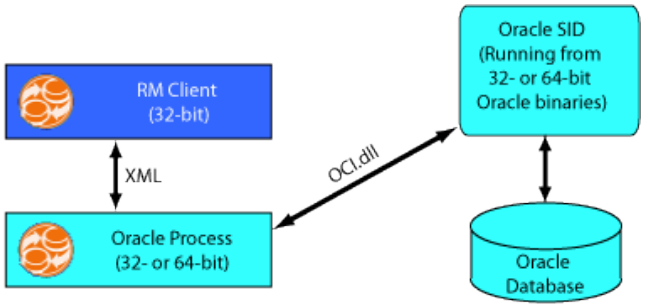

Figure 3. Replication Manager client and oracle_process.exe on Windows

The UNIX model is similar except both the Replication Manager client and the Oracle process are 32-bit regardless of the Oracle instance being 32- or 64-bit.

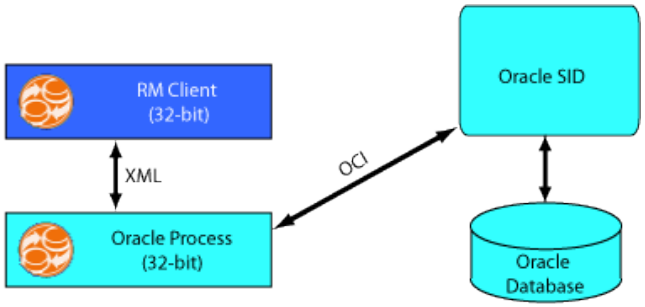

Figure 4. Replication Manager client and oracle_process on UNIX

Retrieving and selecting tablespaces

Once you proceed past the connection panel, Replication Manager spawns an agent process using the operating system username and password collected. After that, Replication Manager connects to the Oracle instance and retrieves a list of tablespaces.

Note: After the initial connection, the Replication Manager Oracle agent detects if Oracle ASM is relevant for the selected Oracle database. If so, it prompts for additional ASM credentials.

The agent retrieves the list of tablespaces under the selected database using a query on standard system views such as dba_tablespaces or dba_data_files.

Notice that certain tablespaces are missing from the list retrieved by the agent. That is because Replication Manager has filtered them out. The following tablespaces are not listed because they are *always* included in every replica and the user cannot deselect them:

- SYSTEM
- OEM_REPOSITORY
- INDX
- USERS

These tablespaces are mandatory for each application set because without them the Oracle instance may not be restarted on the mount host during a mount operation and the replica would be nonviable. Note that the list varies from one Oracle release to another.

Tablespaces that are listed in the Replication Manager Console may be deselected if they are not required in the application set. Replication Manager considers these "user tablespaces" and therefore not required for basic Replication Manager operations.

Once you specify the tablespaces you want to replicate, the application set configuration is complete.

CAUTION: Since Replication Manager is based on array replication technology, it copies data at the LUN level. Therefore, if several tablespaces were created on a single file system for instance, those tablespaces would be included in the replica whether or not they were all selected in the Replication Manager Console. As long as one was selected, the others would implicitly be added to the replica.

If you are interested in a tablespace-level restore, EMC does not recommend you configure an application set in which a subset of the tablespaces share a common file system or volume group. Replication Manager cannot discover this LUN level connection and this would cause problems later during restore operations. Refer to the section entitled *Celerra NFS and Oracle dNFS* for more details. If all the tablespaces will always be restored together, this recommendation is not relevant.

Dynamically discovering and replicating all tablespaces after the application set is created

Oracle application sets that select all the tablespaces presented in the list implicitly follow a design of "automatic discovery" of data objects that are added after the initial application set/job is created. If the database contains 10 tablespaces, and all 10 are selected in the application set, this pattern is communicated to Replication Manager - that the user not only wishes to replicate these 10 tablespaces, but also any subsequent tablespaces that get added to the database. This avoids having to

edit the application set when the database grows. Selecting all the displayed tablespaces automatically enables this "dynamic mode."

Note that if the user deselects one or more tablespaces from the list, Replication Manager then functions in "static" mode whereby it will only discover and replicate the defined set of tablespaces. It will not discover additional tablespaces. However, at runtime, it will still discover new datafiles that were added to the selected tablespaces, to preserve tablespace integrity.

Excluding temporary tablespaces from the replication

Replication Manager 5.3.1 has changed a key behavior related to temporary tablespaces. Prior to version 5.3.1, the behavior was such that temporary tablespaces (tablespaces made of temporary files instead of datafiles) automatically got added to the replica during each replication, regardless of what the user selected in the Replication Manager Console.

It was found that this behavior was not desirable in certain situations. Indeed, temporary tablespaces can be highly volatile files that generate a lot of I/Os and do not contain any real valuable data, and it is best to exclude these entities from the replication process. Replication Manager 5.3.1 enables the user to deselect the temporary tablespaces from the list that is displayed in the application set creation screen.

Dynamically discovering all tablespaces after the application set is created, except for temporary tablespaces

The disadvantage of the change in behavior related to the temporary tablespaces is that the dynamic discovery of the new tablespaces (dynamic mode) cannot be enabled since deselecting one tablespace or more automatically switches Replication Manager to static mode.

To preserve the dynamic mode, while still excluding the temporary tablespaces, do the following:

Set the following environment variable - EMC_ERM_DYNAMIC_TBS_NO_TEMP - to true; this will skip all current and future temporary tablespaces. While defining the application set, select all tablespaces (even the temporary ones); this will preserve the dynamic mode.

In this hybrid mode, new data tablespaces will get discovered and replicated while temporary tablespaces will not.

Creating jobs

Once an application set is defined, create a job that describes a set of options and parameters dictating how Replication Manager should create, store, mount, unmount, and expire the replica.

Defining a mount and/or unmount operation is an optional step when creating the job. Mount and unmount operations can also be performed separately, on demand. Refer to *Oracle mount options* for more information about how to configure Oracle mounts.

Replication Manager Job Wizard

The Replication Manager Job Wizard allows users to define both basic and advanced replication settings. The advanced replication settings are directly related to the application settings (for example, Oracle).

Array level consistent split and application level consistency options

The storage array technology used to store the Oracle database typically does not affect how Replication Manager performs the Oracle operations from an application point of view, except when it comes to data consistency. Both Symmetrix® and CLARiiON® storage arrays provide a "consistent split" feature, allowing a single set of LUNs to be split instantaneously, guaranteeing I/O consistency at the hardware level. Replication Manager provides an option to leverage that functionality if desired.

The Oracle database itself can be quiesced in two major ways:

- Using Oracle's hot backup mode (hot backup)
- Shutting down the database (cold backup)

The array consistency method affects how Replication Manager quiesces and gathers the Oracle data, especially the control files and the online redo log files. The control files and online redo log files are highly volatile I/O bound components of the Oracle database. Tablespaces (and their underlying datafiles) can use Oracle's hot backup mode to ensure recoverability during the split of the mirrors, regardless of the hardware consistency. There is no such mechanism for quiescing the live control files or online redo logs.

For this reason, a replication can only safely split the LUNs containing the control files and online redo log files individually if that replication is using the array-based consistent split option. When the array is using consistent split, the I/Os are frozen at the hardware level, ensuring the consistency of the whole replica (datafiles, control files, and online redo log files as well as any additional objects contained in the application set).

If the replication option is not using the consistent split method, then Replication Manager uses an alternate method to obtain a copy of the control file and the online redo logs, as it cannot safely split the LUNs containing these critical components. In this case, Replication Manager requests a "backup control file" from the Oracle database to obtain an "offline" version of the live current control file. Replication

Manager subsequently performs a series of log file switches to obtain offline (archived) versions of the online redo logs required to recover the data at mount or restore time. The backup control file and archive logs are *copied* to the Replication Manager server as additional files that become part of the replica.

Choosing the proper cosistency settings

So why should you choose one consistency option or another? This section describes each consistency method and provides some guidance regarding why you should choose one option over another.

Consistent split without hot backup mode (repurposing/cloning)

This option relies exclusively on array consistency (consistent split) to obtain a crash consistent copy of the database. This method does not impact the performance of the production database. Hot backup mode can impact performance in highly volatile or large Oracle environments. However, using consistent split without hot backup mode means that the resulting replica cannot be rolled forward

Replica contents	All the major Oracle components (datafiles, online redo logs, control files) are replicated. All of these components must reside on array storage.
Replica usage	This method provides a replica similar to an Oracle database that has been restarted after a power failure. It can be used for crash restart only. Archive logs cannot be applied to it. “What you see is what you get” is another way to describe the resulting replica. Therefore, such replicas are mostly used for repurposing and testing on a separate mount host or cloning a database for use in another application.
Current Ctrl file	Located on the mounted file-system/raw device/ASM diskgroup.
Redo logs	Located on the mounted file-system/raw device/ASM diskgroup.
RM captured archive logs	N/A (no archive logs captured without hot backup mode).

Consistent split with hot backup mode (physical layout backup)

This roll-forward capable option combines the database consistency (hot backup mode) and the array consistency (consistent split). Replication Manager puts the database in hot backup mode, providing a valid backup with the same physical layout as the production database.

Replica contents	Replica includes the online redo log devices and control file devices.
Replica usage	Such replicas provide a valid backup with the same physical layout as the production database.
Current Ctrl file	Located on the mounted file-system/raw device/ASM diskgroup.
Redo logs	Located on the mounted file-system/raw device/ASM diskgroup.
RM captured archive logs	Copied to ‹ERM_TEMP_BASE›/‹sid›/arch/.

Non-consistent split using hot backup mode (data backup)

This roll-forward capable option does not use consistent split but does use hot backup mode. Often referred to as *backups*, this option relies exclusively on the hot backup mode functionality of the database to provide consistency. The online redo logs and control file devices are not added to the replica. Thus, while data contained in the database is accurately captured, the physical layout of the database is not the same as that on the production host.

Replica contents	The replica consists of the datafile LUNs, a backup control file (captured separately as a network file), and a series of archived log files representing the window during which the database was placed in hot backup mode. These replicas are "logical" images of the original database(s), which do not contain the devices for the control files and online redo logs files, therefore, they do not represent a LUN-for-LUN physical copy of the environment but they include the same "logical data," the backup control file and required archive logs necessary to recover the data.
Replica usage	Such replicas are suitable for cases where a backup is required, but the physical layout is not a top priority and only the datafiles need to be split at the array level.
Backup Ctrl file	Located in ‹ERM_TEMP_BASE›/‹sid›/ctrl/.
Redo logs	Not replicated or imported. They are however re-created on the mount host if a read/write recovery is performed on the database.
RM captured archive logs	Copied to ‹ERM_TEMP_BASE›/‹sid›/arch/.

Offline with consistent split (cold backup —not available for non-SYSDBA users)

This roll-forward capable option uses the array consistency (consistent split) and also takes the database or selected tablespaces offline during the replication. This method produces a cold backup of the database with the same physical layout as the production database, even though the online redo log devices do not contain any

usable information (since the database was shut down cleanly). This option is rarely used.

Replica contents	Replica includes the online redo log devices and control file devices.
Replica usage	Cold backups are not the most commonly used method as per their offline nature but the option is available nonetheless.
Current Control file	Located on the mounted file-system/raw device/ASM diskgroup.
Redo logs	Located on the mounted file-system/raw device/ASM diskgroup.
RM captured archive logs	N/A (no archive logs captured in cold backups).

Offline without consistent split (cold backup —not available for non-SYSDBA users)	
Description	This roll-forward capable option shuts down the database (or places individual tablespaces offline if only a subset of the tablespaces had been selected). This option is rarely used.
Replica contents	Does not include the online redo log or control file devices. The option provides a cold copy of the datafiles on the target device, and a captured control file (network transferred).
Replica usage	Cold backups are not the most commonly used method as per their offline nature but the option is available nonetheless. This option is used when the layout likeness of the database is not a priority and the database can be taken offline.
Backup Ctrl file	Located in ‹ERM_TEMP_BASE›/‹sid›/ctrl/.
Redo logs	Not replicated or imported.
RM captured archive logs	N/A (no archive logs captured without hot backup mode).

Online without hot backup mode without consistent split (third-party backup integration)	
Description	The ability to skip the hot backup mode for online replications is now available for arrays that do not have the consistent split capability. The user is responsible for putting the database in and out of hot backup mode outside of Replication Manager. Mount options that automatically recover the database are not guaranteed to work with replicas taken in this mode since Replication Manager is not aware of any archive logs or backup control file created during the

	replication.
Replica contents	Does not include the online redo log or control file devices. The option provides a backup of the datafiles on the target device.
Replica usage	These replicas can integrate a third-party backup product with Replication Manager in such a way as to allow it to manage the hot backup mode on its own. This is typically the case for solutions where Replication Manager is not used to recover the database, but it mainly leveraged for the array replication capability. Replication Manager does not perform capture of the archive logs, backup control files, or recovery management during a mount of this type of replica. Similarly, Replication Manager restores of this type of replica do not include backup control files or archive logs.
Backup Ctrl file	Not captured. The third-party backup product is responsible for capturing that.
Redo logs	Not replicated or imported.
RM captured archive logs	The third-party backup product is responsible for capturing the archive logs generated during the hot backup period.

Additional optional files and procedures

Additional options can be used to further customize the replication job.

Capturing the initialization file

Replication Manager copies the initialization file, by transferring it across the network to the Replication Manager server during the replication process. The initialization file can be used later during the mount operation, to restart the instance. If necessary, the file can be altered to facilitate changes that have occurred to the environment, for example, if the paths to certain components like the control files have changed.

Replication Manager version 5.3.1 supports restoring the initialization file. The file is restored, with a ".restored" extension, to the ERM_TEMP_BASE location, during the restore of a replica.

Oracle *9i* introduced the spfile, which is a binary version of the text-based initialization file. In order to preserve these files, Replication Manager needs to know

which of the two file types is used by the database(s) targeted for the replica. When Replication Manager creates a new job, the Oracle agent attempts to discover if the spfile is being used. If that is the case, the location to the spfile is known to the Oracle instance and therefore is known to Replication Manager. Replication Manager displays the path to the spfile in the console for verification. If Replication Manager finds that the spfile is not used, the default Oracle initialization file location is used instead and in this case the user must verify that the path is correct, as the initialization file is not known to the Oracle instance once it has started.

Capturing the archive log directory

In addition to the tablespaces replicated by the job, and possibly the control files and redo log devices (if the consistent split option is used), there is an option to include the archive log directory and the Flash Recovery Area device as part of the replica (in Oracle 10*g* environments and above).

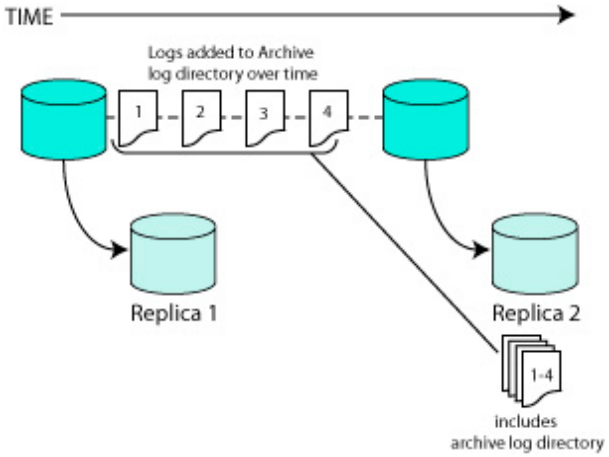

Figure 5. Including the Oracle archive log directory

Including the archive log directory is different from Replication Manager's capture of archive logs during a hot backup-based replication and does not apply to the current replica. The device containing the archive logs will be split at the same time as the datafiles. The archive log directory is only relevant to replicas spawned previously from the same job. The archive log directory can be applied to an earlier replica to roll it forward. Refer to Figure 5.

The archive log directory can be configured in the following ways:

```
log_archive_dest = '/filesystem/'
```
(typical in Oracle Standard Edition)

```
log_archive_dest_1 = 'location=/filesystem/'
```

(typical in Oracle Enterprise Edition)

```
log_archive_dest_1 = 'location=USE_DB_RECOVERY_FILE_DEST'
```

(10g and up)

```
log_archive_dest_1 = 'location=+ASMDG'
```

All of these types of archive log directory are supported by Replication Manager.

Multiple archive log destinations may be configured on the database server. Replication Manager only considers one for replication, as well as a source from which to capture archive logs. Replication Manager will follow this logic to determine which location to choose:

If one location is using the USE_DB_RECOVERY_FILE_DEST keyword, Replication Manager gives preference to that one.

Otherwise, it will consider the first location that is found as PRIMARY, VALID and not used as part of a standby (data guard) configuration.

Other locations will be ignored.

Using pre- and postreplication scripts

Replication Manager also allows the user to specify pre- and postreplication scripts to be run, respectively, before and after the split of the devices. These optional scripts can be written in any kind of language and perform tasks such as disconnecting applications from the Oracle database, terminating batch jobs attached to the database, reconnecting applications to the database, and other tasks.

The only minimum requirements, to which the scripts must conform, for hot backup-based replicas, are:

- **Prereplication scripts** — Ensure that the tablespaces involved in the replication are put in hot backup mode. This task is normally performed by Replication Manager if no prereplication script is specified.
- **Postreplication scripts** — Ensure that the tablespaces involved in the replication are taken out of hot backup mode. This task is normally performed by Replication Manager if no postreplication script is specified.

For offline replicas, the minimum requirements to which the scripts must conform are:

- **Prereplication scripts** — Ensure that the database is shut down (or individual tablespaces taken offline). This task is normally performed by Replication Manager if no prereplication script is specified.
- **Postreplication scripts** — Ensure that the database is restarted (or individual tablespaces put online). This task is normally performed by Replication Manager if no postreplication script is specified.

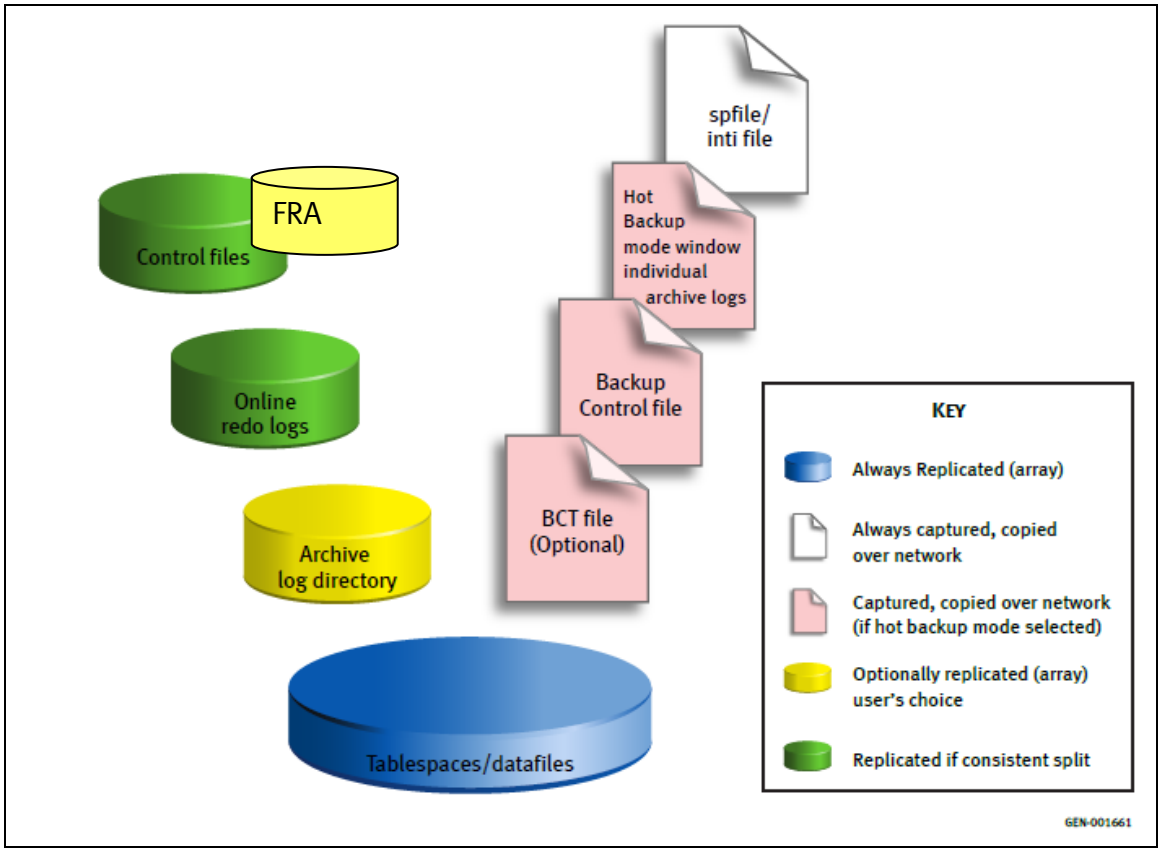

Figure 6. What pieces get replicated or copied in which cases

Flash Recovery Area (FRA)

Replication Manager can also replicate the Flash Recovery Area when configured to do so.

The Flash Recovery Area is a database component introduced in Oracle 10*g*. Oracle documentation describes extensively how to use and configure this component. As far as Replication Manager is concerned, it can be summarized as a multipurpose container that can hold, among other things:

- Flashback logs. These logs contain the undo information that the Oracle Flashback feature relies on.
- Archive logs. If the archive log directory is specified with the USE_DB_RECOVERY_FILE_DEST keyword, Oracle effectively manages the archive logs and uses the Flash Recovery Area to store them.
- RMAN catalog related information and backup pieces.

The Flash Recovery Area can be specified as a file system or an ASM diskgroup. Replication Manager works with both types of Flash Recovery Areas; however, if the Flash Recovery Area is file system-based, it needs to be one of the supported file system types for Replication Manager on that operating system.

Note: Replicating the Flash Recovery Area is optional but if you choose to replicate the Flash Recovery Area, it must reside within a supported storage environment.

Block Change Tracking file (BCT file)

Oracle 10g introduced the concept of BCT file (Block Change Tracking) in the form of a relatively small binary file that tracks the changed blocks of the database in a bitmap. By making use of this file, RMAN incremental backups benefit from a significant performance boost by virtue of the fact that RMAN does not have to scan all data blocks to find if they have changed or not (and so whether or not they need to be backed up). It will consult the BCT file instead and know instantly if the data block can be skipped.

If BCT is enabled on the production database, Replication Manager 5.3.1 and later can leverage this by copying the file as part of the replication taking place (the **Copy BCT file** option appears on the advanced replication options panel of the Replication Manager Console). When the replica is mounted to a host with the "catalog with RMAN" option, the BCT file will be retrieved from the Replication Manager server and placed on the mount host in the ERM_TEMP_BASE/‹sid› location. Any incremental backups invoked through RMAN at that point will automatically make use of the BCT file.

The BCT file is composed of eight slots, each representing a bitmap of the data blocks since the last backup or the last switch between slots. For the copy of the BCT file to be useful, it needs to be aligned with the last backup slot. Consequently, Replication

Manager will invoke an Oracle provided *dbms_backup_restore.bctswitch()* stored procedure to perform a *switch to the next slot* operation before copying the BCT file.

The BCT file has no purpose in a restore scenario, therefore it is not restored, alongside of the initialization file or backup controlfile, during the restore of a replica.

For details related to the BCT file and mount, see the *Integration with Oracle Recovery Manager (RMAN)* section.

Running the job

The previous section explained the options available when creating a job. This section summarizes the key steps involved in running the job.

Full discovery

When a job starts, Replication Manager performs a full discovery of the tablespaces associated with the corresponding application set. Replication Manager uses various OCI calls to map the datafiles that constitute the tablespaces that are part of the application set.

The datafiles may be stored in any of the following formats:

- UNIX filesystems
- Windows NTFS
- Raw volumes (third-party LVM or native operating system)
- Raw disks
- ASM volume groups
- Network File Systems (Linux, Solaris, and HP platforms only)

For consistent split replicas, Replication Manager also discovers the control file devices and online redo logs. If the archive log directory has been included in the job, it will be discovered as well. Note that Replication Manager uses a specific selection criteria to determine which archive log location to pick, if several are defined.

The resulting objects are processed by the storage services agent, which determines what storage containers (for example, which LUNs) hold these objects and based on that information, what target storage is suitable to contain the replica.

Log switches and hot backup mode

If hot backup mode is selected (depending on the options defined in the job), Replication Manager performs the following steps:

1. The Oracle agent queries standard Oracle views using OCI to determine the current log sequence number.
2. Replication Manager instructs Oracle to perform a log switch on the database to archive the current log sequence using the SQL*Plus command line.

3. Prior to version 5.3, Replication Manager put each tablespace individually into hot backup mode. Starting with Replication Manager 5.3, the Oracle agent places the entire database in hot backup mode via SQL*Plus command line. If individual tablespaces are selected for replication, then Replication Manager places selective tablespaces individually in hot backup mode.
4. Replication Manager splits the source and target LUNs.
5. Prior to version 5.3, Replication Manager released each tablespace individually out of hot backup mode. Starting with Replication Manager 5.3, the Oracle agent releases the entire database from hot backup mode via SQL*Plus command line. If individual tablespaces were selected for replication, then Replication Manager releases selective tablespaces individually in hot backup mode.
6. Replication Manager instructs Oracle to perform a log switch on the database to archive the current log sequence using the SQL*Plus command line. The begin and end backup markers will now be accessible in the archived version of the online redo logs.
7. Replication Manager requests a backup control file using the SQL*Plus command line.
8. The Oracle agent queries standard Oracle views using OCI to determine the current log sequence number.
9. Using the sequence numbers obtained at steps 1 and 8, Replication Manager computes the list of archive logs that cover the entire time that the database was placed in hot backup mode. These are the archive logs required to recover the database at a later time.
10. If **Copy BCT file** was selected, invoke the *dbms_backup_restore.bctswitch()* procedure to switch to the next internal incremental backup slot in the BCT file. Then copy the BCT file to the temp area, to be transferred to the Replication Manager server at a later time.

Network transfer and cataloging

The last important phase of the replication is cataloging the replica. Replication Manager catalogs the following data:

- Metadata describing the contents of the replica
- Archive logs captured during the replication (if applicable)
- Backup control file (if applicable)
- spfile/initialization file
- BCT file (if selected)

These files are captured and transferred to the Replication Manager server in this final phase of the replication.

Note: The Oracle agent uses a temporary location (referred to as ERM_TEMP_BASE) to store files relevant to a replication while it is taking place. It uses that to centralize the files needed, before sending them to the server. The files in the temp area are deleted at the end of the replication.

When the archive logs are stored in an ASM diskgroup, the Oracle agent instructs the RMAN utility to make a copy of the selected archive logs from the ASM diskgroup to the temporary location, so that the Replication Manager agent can transfer them to the Replication Manager server and catalog them.

The ERM_TEMP_BASE location defaults to /tmp on UNIX systems or the system-wide temp location on Windows. It can be overridden by setting the ERM_TEMP_BASE environment variable to another location.

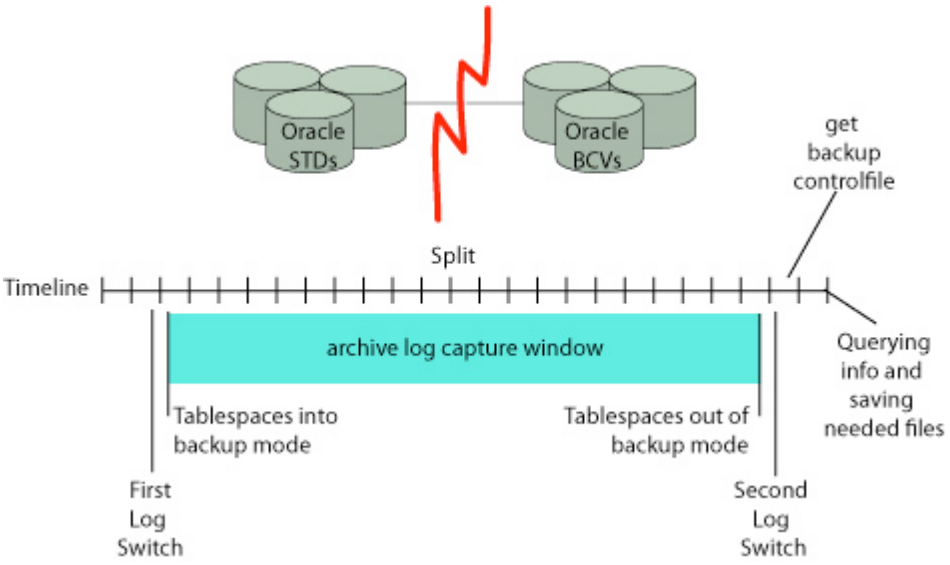

Figure 7. Hot backup mode timeline

Replication of the Flash Recovery Area (FRA)

If the FRA was selected as an object to include in the replica, it will be discovered as well and replicated at the array level like the datafiles. Note that if you need to replicate the contents of the FRA, EMC recommends that you use the consistent split option, since it is the only one that guarantees I/O consistency. The FRA does not have the equivalent of the "hot backup mode" that the tablespaces have, therefore there is no application-level way of quiescing it. Consistent split ensures the FRA will be crash consistent.

This matters especially if the intent of the replica is to use the flashback feature when the replica is mounted or restored. The validity of the flashback logs contained in the FRA then becomes critical.

The FRA and the archive log directory can be associated with each other using the USE_DB_RECOVERY_FILE_DEST keyword as a value to any of the log_archive_dest_x parameters. The FRA contains the archive log directory. Therefore they should either both be selected or unselected in the replication job.

If they are configured independently (db_recovery_file_dest pointing to one location, and log_archive_dest pointing to another), they can be selected/unselected independently, if their underlying storage layout is distinct (comprised of different file systems/volume groups/diskgroups).

Support for RAC awareness during replication of RAC databases

Replication Manager 5.4 added support for cluster awareness during replication of RAC databases. During replication, if the node used for application set creation is not accessible, Replication Manager runs the replication on another node in the RAC. This feature can be accessed through the IP used to register the client, and does not have a GUI option.

Pre-requisites for RAC awareness:

- Replication Manager client should be installed on all nodes of the RAC.
- Replication Manager client should be registered using
  - VCS Failover IP or with the IP resource configured in Oracle service group for the database or Oracle RAC VIP managed by VCS in case of SFRAC.
  - Oracle RAC VIP/ Failover/Cluster IP in case of other RACs
- The database user, i.e., SYSDBA privileged user credentials should be the same for all DB instances running in the RAC. All Oracle password files across all production RAC nodes should be in sync.
- RAC-aware feature will work only if the node failover occurs prior to running the job, that is, prior to start of replication or start of restore. If failover happens in the midst of running replication or restore, the operation fails.
- In Advanced Replication settings, select SP file in the Copy parameter file to RM Server option. The SPfile can be a global SPfile on shared location and not referencing the local SID (in the file name) or the SPfile of a RAC instance running on the node selected for job creation. Replication Manager does not support use of init file in Copy parameter file to RM Server option.
- Replication Manager does not support OMF for database Creation of non-ASM databases

At the start of replication, RM now determines the RAC Database instance and ASM instance (for ASM-RAC databases) running on the node instead of retrieving the values of the RAC database instance and ASM instance that were saved during application set creation. So, if a failover has happened after application set creation, the values of RAC database instance and ASM instance(for ASM-RAC databases) would be different now, but by determining the values again dynamically, RM connects to the right instances.

At the end of Replication, RM catalogs the RAC database instance that it dynamically discovered. This is used as the database SID for mounts without using database rename.

Support for fail-over standalone databases in a clustered environment

In addition to supporting RAC databases, Replication Manager 5.4 added support for standalone databases that are configured to failover from node to node in the cluster. During replication, if the node that was used for application set creation is not accessible, Replication Manager runs the replication on another node in the cluster.

Configuration pre-requisites:

1. Replication Manager client should be installed on all nodes of the cluster.
2. Replication Manager client should be registered using
   - VCS Failover IP or with the IP resource configured in Oracle service group for the database
   - VIP/ Failover/Cluster IP
3. The oratab file should have an entry for all possible SIDs that can run on the given node (passive and active instances)
4. The tnsnames.ora on all the nodes that are candidates for a fail-over should have an entry for the standalone database corresponding to that on the node owning the database.
5. The following files should be accessible to all nodes of the cluster where the database can run:
   - Database init/spfile
   - Password file

Note: Create the files on a shared filesystem or manually copy the files to their corresponding location on all nodes ($ORACLE_HOME/dbs).
The dump directory location specified in the init file must exist on all nodes in the cluster.

---

6. The feature will work only if the node failover occurs prior to running the job, that is, prior to start of replication or start of restore. If failover happens in the midst of running replication or restore, the operation fails.

Support for Oracle 11g R2 RAC One Node Feature

Replication Manager 5.4 onwards supports Oracle 11g R2 RAC One Node feature on Linux. Replication Manager is RAC aware for such databases. The pre-requisites for supporting this feature are

1. Replication Manager client should be installed on all the nodes of the RAC.
2. Register the host using RAC VIP of the node that currently owns the RAC One Node database.

Note: SCAN IP cannot be used for registering the host.

---

3. In case of a failover, the RAC VIP and the database should come up on the same RAC node. In case, the RAC VIP and the instance come up on different nodes of the

cluster, the user needs to migrate the RAC VIP to the appropriate node using "ifconfig" command. For example:-

a. If Omotion is used to migrate the instance then on the node previously owning the RAC VIP run command similar to:
```
ifconfig eth0:<n> <RAC VIP> netmask <Netmask> down
```

b. On the RAC node where the database has come up after a failover run command similar to:
```
ifconfig eth0:<n> <RAC VIP> netmask <Netmask> up
```

c. Check the node now owning the RAC VIP resource in the output of crs_stat and confirm that it is pointing to the correct node.

4. If failover happens in the midst of running replication or restore, the operation fails. Node failover should occur prior to running the job, that is, prior to start of replication or start of restore.

5. The **tnsnames.ora** file on all the nodes that are candidates for failover must contain an entry for all the possible RAC One node database instances. For example, Let us consider a two node RAC configuration with node names as node1 and node2, and the RAC One Node database name as racone. Let us assume that the database is initially running on node1 as racone_1. In case a failover happens via Omotion to node2, the instance would be renamed to racone_2. However, in case node1 goes down, the instance name remains as racone_1 when it comes up on node2. Consequently, the tnsnames.ora file on both the nodes should have entries for both the database instances. Sample entries are given below:-

```
RACONE_1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = SCAN_IP)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = racone)
      (INSTANCE_NAME = racone_1)
    )
  )


RACONE_2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = SCAN_IP)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = racone)
      (INSTANCE_NAME = racone_2)
    )
  )
```

6. Mount as Real Application Cluster is not supported for RAC One Node database replicas.

7. In Advanced Replication settings, select SP file in the Copy parameter file to RM Server option. The SPfile can be a global SPfile on shared location and not

referencing the local SID (RAC instance) or the SPfile of a RAC instance running on the node selected for job creation. Replication Manager does not support use of init file in Copy parameter file to RM Server option.

Automatic Storage Management

Automatic Storage Management (ASM) is a logical volume manager introduced as a feature of Oracle 10*g*. It provides a diskgroup capability for storing Oracle database components. This section of the paper describes how Replication Manager interacts with ASM. In addition to the information provided here, other sections of this paper provide more details about how ASM interacts with Replication Manager restores, mounts, and RAC environments.

Interactions with the production host ASM instance

Just like the database instance manages the database, there is a specialized ASM instance that manages the ASM diskgroups. So, in configurations that include ASM, Replication Manager must communicate to both the database and ASM instances. Replication Manager communicates with the ASM instance using the same tools that it uses otherwise. OCI calls run queries against common ASM views such as v$asm_disk and v$asm_diskgroup, and SQL*Plus internal scripts perform operations such as dismounting and remounting certain diskgroups.

The section *ASM connection and authentication* describes the extra set of credentials that are required for Replication Manager to communicate with the ASM instance. From there, Replication Manager has the tools to replicate, mount, and restore databases built on ASM diskgroups.

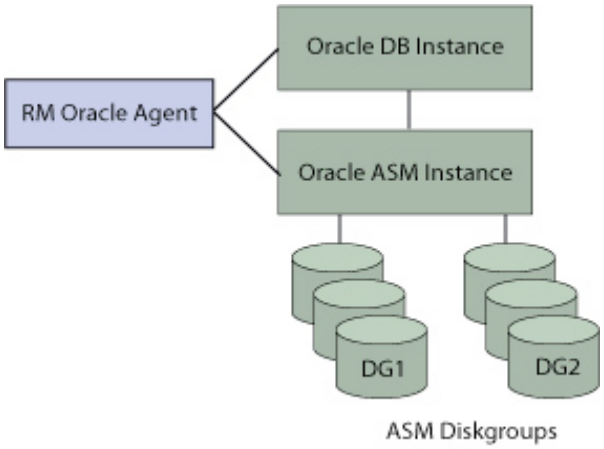

Figure 8. The Replication Manager Oracle agent interacts with both database and ASM instances when ASM diskgroups are involved

Using an alternate ORACLE_HOME for ASM

Replication Manager version 5.2 and later support the use of two separate ORACLE_HOME directories, one for ASM instances and one for Oracle databases if you follow these guidelines:

Prior to version 5.2.3, Replication Manager required the same Oracle operating system user and group for both the database ORACLE_HOME and ASM ORACLE_HOME. Replication Manager version 5.2.3 and later support a separate operating system user and group to access the database ORACLE_HOME and ASM ORACLE_HOME on UNIX and Linux platforms.

The environment must list the ASM instance explicitly in the /etc/oratab file.

On Linux platforms, if you plan to mount a Real Application Cluster, dual home environments are allowed. Prior to version 5.3.1, the OS user on the target RAC had to be the same for ASM and Oracle databases. With version 5.3.1 and later, this restriction has been removed.

Limitations

There is a list of special considerations to keep in mind when implementing Replication Manager in ASM environments.

Note: Replication Manager does not support ASM on Windows platforms at this time.

Raw disks only on UNIX platforms

Replication Manager supports the replication and handling of ASM diskgroups when those diskgroups are created using exclusively raw disks. The use of raw volumes managed by third-party or native LVMs as source disks to create ASM diskgroups is not supported by Replication Manager. For Linux platforms specifically, the use of *ASMlib disks* is supported and recommended as of Replication Manager 5.1 SP2.

No character device file

Replication Manager does not support the use of character files created manually using commands such as mknod as a means to identify a disk and subsequently map ASM diskgroups to that disk. An example is /usr/asm/mydisks/disk1 where disk1 is a character file defining a disk by using its minor and major number. This method is not compatible with the current model that Replication Manager uses to map ASM diskgroups.

```
# cd /usr/asm/mydisks
# mknod disk1 c 32 20
# ls -l /usr/asm/mydisks
crw-r--r-- 1 root other 32, 20 May 7 07:50 disk1
```

The use of the ASMlib driver and creation of ASMlib disks (that is. ORCL:VOL1) to use

with ASM is supported and recommended as of Replication Manager 5.1 SP2. The use of raw disks with ASM in the following way is also supported.

For example on Linux: /dev/raw/raw1

Or on Solaris: /dev/rdsk/c5t6d36s6

ASMlib driver support on Linux platforms

ASMlib is a driver for the Linux platforms that allows user-friendly configuration and management of volumes that can be presented to ASM on which it can create diskgroups. Instead of binding character raw devices to block devices like /dev/sdb1, use ASMlib to create ORCL:MYVOL1 on /dev/sdb1 and have ASM address this disk as such.

ASMlib version 2.0 and later come with an "oracleasm" script that, given the corresponding arguments, can perform operations such as listing ASMlib volumes, labeling new ASMlib volumes, renaming volumes, querying for the corresponding block device from an ASMlib volume, and so on. Replication Manager has integrated calls to that aforementioned script to perform basic operations that must occur to allow replications, restores, and mounts of Oracle databases built on ASM diskgroups using the ASMlib driver (Replication Manager 5.1 SP2).

The following sections give additional details on various aspects of the ASMlib integration:

ASMlib volumes are renamed during mount

ASMlib volumes are renamed during restore

Production host ASMlib volumes clobbering

External mirroring

Because Replication Manager is tightly integrated with the array replication technologies and those rely on a precise understanding of which LUNs are involved in a particular replication operation, Replication Manager cannot support ASM diskgroups that use normal or high redundancy.

Active ASM disks within a diskgroup can be very volatile in setups where normal or high redundancy is used. That makes it difficult for Replication Manager to replicate those diskgroups as units.

Figure 9 shows an ASM diskgroup made of disk1, disk2, and disk3. The diskgroup is using normal redundancy, which means there is one mirror disk for each disk. In this scenario, disk2 has failed and disk2' is rebalancing its extents with disk1 and disk3. The diskgroup at this point is actively represented by disk 1, disk2', and disk3. Replication Manager cannot support this configuration because it would be replicating the diskgroup at the disk level and in this case targeting disk1, 2 and 3, thus creating a bad copy of the ASM diskgroup since disk 2 is suddenly faulty.

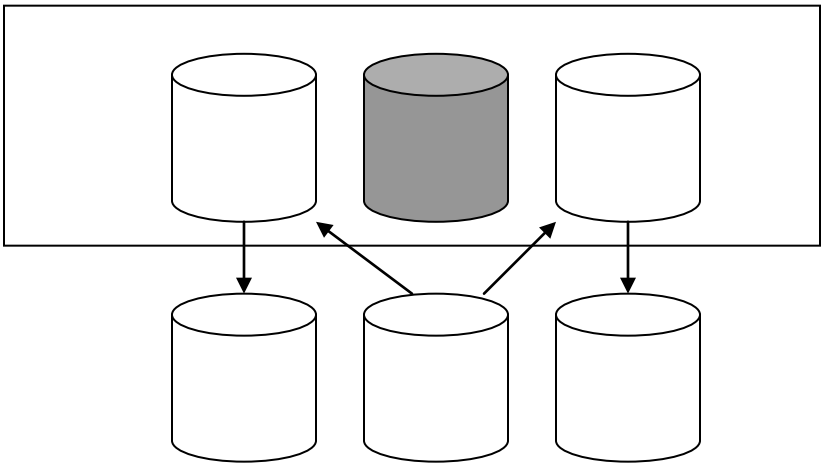

Figure 9. Normal redundancy ASM diskgroup with one failed disk

Replication of ASM diskgroups

This section describes issues to be aware of when replicating ASM diskgroups using Replication Manager.

Discovering ASM diskgroups

When the database is stored on ASM diskgroups, the Oracle agent performs an additional mapping in order to decompose a diskgroup into disks that can be replicated. It uses OCI to query the ASM instance to map the diskgroups to the disks. For example, on Linux, +DG1 might map to ORCL:Vol1, ORCL:Vol2 and ORCL:Vol3. These volumes further correspond to LUNs on the array (one-to-one mapping).

Rebalancing issues

Rebalancing is a very important ASM feature that consists of constantly redistributing data extents within a given ASM diskgroup, across all the disks of the diskgroup, in order to optimize load on the different disks.

Replications of ASM-based Oracle databases should use the consistent split option. Thanks to the array level I/O consistency, the rebalancing of extents from one disk to another never causes an inconsistency within a diskgroup as all the I/Os will be frozen at the same time during a split operation.

When not using the consistent split option, the Replication Manager Oracle agent performs an additional step to counter the fact that I/O consistency is not ensured at the array level. The agent sets the rebalancing power to 0 before the split and monitors the rebalancing activity surrounding the involved diskgroups. Replication proceeds when the rebalancing activity has stopped, the non-consistent split occurs across the disks of the diskgoup, and the rebalancing power gets reset to the default after the split is complete.

While this greatly minimizes the chances of I/O consistency problems, the non-consistent split solution is not guaranteed to maintain consistency under all

circumstances, as a rebalancing operation may occur anyway, when initiated outside of Replication Manager, or if a disk is added to the diskgroup. For this reason, the consistent split method is highly recommended.

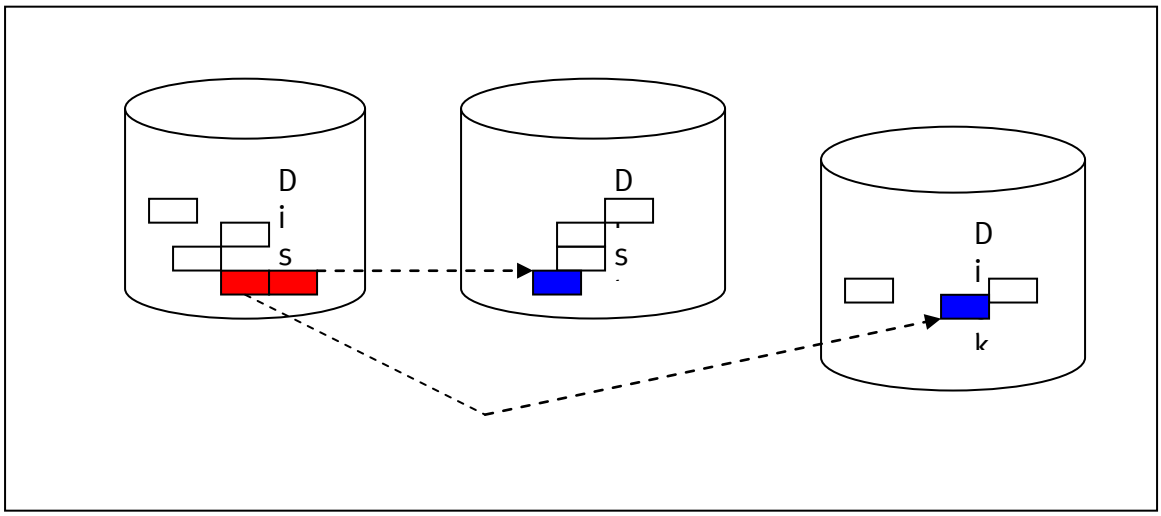

Figure 10. ASM diskgroup with data extents being redistributed from disk1 to disk2 to disk3

Celerra NFS and Oracle dNFS

Replication Manager 5.2.2 introduced support for Celerra NFS SnapSure™ snapshots and Celerra Open Replicator snapshots. Oracle agent behavior is largely transparent to the user, aside from mount option notes (see the *Mounting an Oracle replica* section). From a replication standpoint, the behavior of the Oracle agent is similar to Symmetrix- and CLARiiON-based replications using the non-consistent split option.

Oracle Direct NFS (dNFS) I/O mode lists the various NFS exports available to the host as well as up to four network paths that can be used to access them. The use of Oracle dNFS I/O mode, which is available with Oracle 11g R1 and later, does not affect Replication Manager operation. If oranfstab is used, Replication Manager is unaffected, because Replication Manager maps network filesystem exports through the /etc/fstab file.

Note: Replication Manager ignores Direct NFS on the production and mount hosts. In other words, it does not manipulate any existing oranfstab file on the mount host, nor does it preclude the user from having one.

Restoring an Oracle replica

This section describes issues to remember related to the restore of Oracle replicas.

Restoring Oracle replicas with Replication Manager

Just as Replication Manager allows you to create replicas of Oracle databases (full database replicas or a subset of the tablespaces making up that database), it also allows you to restore those replicas. The basic restore operations are:

1. Put the database, or part of the database, in the proper state to be restored.
2. Dismount/deport the underlying storage layer.
3. Perform the actual restore of the data (for example, reverse sync a CLARiiON SnapView™ clone).
4. Import/remount the underlying storage layer.

The restore operation fails if the database user who created the replica does not have SYSDBA privileges.

Full versus individual tablespace restore

On restore operations, Replication Manager offers a choice between restoring the entire replica, including all subcomponents, or restoring only individual tablespaces. If the online redo log and the control file devices were part of the replica (consistent split), they can be selected for restore as well. The same applies to the archive log directory and the Flash Recovery Area (FRA), if it is part of the replica.

The lowest level of granularity that can be selected for a restore is tablespace. The Replication Manager Console allows you to browse the datafiles but cannot allow individual restore of datafiles because tablespaces are handled as a logical backup unit for Replication Manager.

The entire set of control files can be restored as a unit. You cannot select one particular control file to restore. Similarly, the entire set of online redo logs can be restored as a unit. You cannot select just one online redo log. Further restrictions apply depending on the affected entities. The section *Affected entities* has more information.

During a restore operation, Replication Manager searches for the target production Oracle database. If that production database is online and can be connected at the time of restore, it attempts to place it in an appropriate state as follows:

- If the replica contains the entire database and the entire database is selected to be restored, Replication Manager attempts to shut down the database.
- If the replica does not contain the entire database or if it contains the entire database but only a subset of the tablespaces are selected for the restore, Replication Manager attempts to place the affected tablespaces offline.

Note: Tablespaces such as "SYSTEM" cannot be placed offline individually. For that reason, restores involving the SYSTEM tablespace or other critical core tablespaces, require a database shutdown.

- If the database is already down at the time of restore, Replication Manager performs the restore without changing the state. This would be the case if the database crashed and could not be started.
- If the database cannot be connected but yet is not shut down, Replication Manager fails the restore because it cannot determine the state of the database. In such cases, EMC recommends that the user manually shut down the database and then retry the restore operation.

Affected entities

The Replication Manager Oracle agent 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 replica, Replication Manager ensures that all of the underlying datafiles, file systems, and volume groups are discovered. When a job runs, Replication Manager maps the source data to target devices based on your job selections and replicates the data.

IMPORTANT: The granularity of restore operations is tightly linked to the underlying storage layout on which the database was built.

If physical dependencies are present within those five tablespaces, Replication Manager may not be able to detect it. This may cause problems during restore.

For example, say there are five tablespaces consisting of one datafile each, all of which reside on the same file system. If two tablespaces are selected for restore, the restore cannot complete successfully because the five tablespaces in this case are all sharing the same file system. Since Replication Manager works at the LUN level, Replication Manager attempts to revert the entire LUN, which includes all the tablespaces. The result ranges from failure (unable to dismount the file system since three of the tablespaces are still active) to corruption (if the database is down at the time of restore, the file system will be dismounted, but the desired result is not obtained).

Also keep in mind that even two distinct file systems may be associated if they are built on volumes of a common volume group. Replication Manager replicates 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. Replication Manager can create different kinds of replicas, as described in the section *Creating jobs*.

Restoring consistent split replicas

Replicas created using the consistent split option contain the datafiles composing the selected tablespaces, the control file devices, and the online redo log devices.

When restoring such a replica, it is always advisable not to select the online redo logs or control files, as restoring those devices means overwriting the latest transactions committed to the database before the restore occurred.

However, depending on the situation that led to the restore, it is possible that restoring those critical objects will also be required if, for example, the control files and online redo logs were lost.

If the hot backup mode was not selected, the restored data is as good as a crash consistent copy and can only be restarted from the point in time at which the replica was taken. Roll forward will not be possible. For that reason, EMC recommends that you restore the control file and online redo logs in that specific case. But in general, crash consistent copies of the database without using hot backup mode should not be considered for restore as they do not represent a valid Oracle backup.

Restoring non-consistent split replicas

Replicas created without the consistent split option contain the datafiles composing the selected tablespaces but do not contain the redo logs and control file devices. They do contain a backup control file that was captured through the network during replication. That backup control file is restored in the ERM_TEMP_BASE/‹sid name›/ location and may be used if required although if a current version of the control file is present on the host at the time of the restore, EMC recommends that you use that one instead. Replication Manager 5.4 and later, the backup control file is restored to ERM_TEMP_BASE/‹db name› location.

Restoring online with hot backup replicas

Both consistent split and non-consistent split replicas have an “online with hot backup mode” option. Therefore, despite their differences, they have the following in common:

The restored datafiles are in hot backup mode — and therefore ready to have archive logs applied to them to roll forward (unless a mount was performed on the replica with a recovery option).

The archive logs captured during the replication, covering the time window during which the database was in hot backup mode, are restored to the ERM_TEMP_BASE/‹sid name›/ location. If the archive log directory is still available and contains the appropriate archive logs already, there is no need to use those restored by Replication Manager. In Replication Manager versions 5.4 and later, the backup control file is restored to ERM_TEMP_BASE/‹db name› location. In Replication Manager versions 5.4 and later, the archive logs are restored to ERM_TEMP_BASE/‹db name› location

A backup control file is restored to the ERM_TEMP_BASE/‹sid name›/ location and may be used if required although if a current version of the control file is present on the host at the time of the restore, EMC recommends that you use that instead. In Replication Manager versions 5.4 and later, the backup control file is restored to ERM_TEMP_BASE/‹db name› location.

The initialization file that was captured during replication gets restored as ERM_TEMP_BASE/init‹SID›.restored.

Restoring the archive log directory device

If the optional archive log directory device was added to the replica, by explicitly selecting it in the Advanced Replication Options panel of the Job Wizard, then it is also available for restore.

Note: Consider carefully whether you should restore the archive log directory, because restoring that overwrites the current contents of production archive log directory and rolling forward on the restored copy would become impossible.

Typically, the replica containing the archive log directory would be mounted to an alternate location. From there, selected individual archive logs file can be manually retrieved if a set of archive logs was missing from the production host.

The archive log directory portion of a replica must always be seen as an additional backup copy of the archive log directory that may have value to fill in the gaps for previous replicas.

Restoring the Flash Recovery Area

If the optional Flash Recovery Area device was added to the replica, by explicitly selecting it in the Advanced Replication Options panel of the Job Wizard, then it is also available for restore.

Note: Consider carefully whether you should restore the Flash Recovery Area. Doing so would overwrite the current contents of the Flashback logs. Additionally, the latest archive logs, created after the replica was taken, would also be overwritten if the archive log directory was included in the FRA. That makes it impossible to use the logs to roll forward the restored copy.

If the FRA (containing flashback logs) was replicated using consistent split, a flashback database operation on the restored replica may be possible if the FRA contains sufficient flashback data to complete the desired restore (see Oracle documentation for how to configure the size of the FRA).

Manual recovery after restore

Whether it is after an individual tablespace restore or a full database restore, Replication Manager does not attempt to recover the datafile, tablespace, or database on the production system. Replication Manager restores the selected objects in place and provides the necessary archive logs and control files in order for the administrator to recover the database to at least the point in time at which the given replica was taken. However, because the restore scenarios and recovery requirements may vary greatly on the production system, Replication Manager does not attempt to recover the data.

If the full database was restored, the instance remains in a shutdown state. If individual tablespaces were restored, they remain in an offline state. The database administrator must perform the recovery manually.

ASM diskgroups during restore

Additional operations are performed when a replica involving ASM diskgroups is restored.

Dismounting and optionally renaming ASM diskgroups

After the appropriate database or tablespaces have been taken offline, using SQL*Plus scripts, the Replication Manager Oracle agent dismounts the corresponding ASM diskgroups on which these entities are built.

After Replication Manager restores the LUNs, it checks to see if the replica that was just restored was ever mounted; in which case, the diskgroup rename feature may have been used to alter the names of the ASM diskgroups. In this situation, Replication Manager renames the diskgroups back to their original names. Finally, it remounts the ASM diskgroups.

Note: The same affected entity concepts apply to ASM diskgroups as do to other LVMs or file systems.

Refer to "ASM instance user privileges" for the user account needed for the ASM instance. Replication Manager uses the ASM instance user info provided during application set creation time to perform the ASM diskgroup dismount, remount operations during restore.

For example, if multiple databases are built on one ASM diskgroup called +DG1, but only one was logically replicated with Replication Manager, Replication Manager has no record of the other databases. Therefore, during a restore operation, one of two things can happen:

- Both databases are online. Replication Manager shuts down the database it has replicated but not the other database in the diskgroup. In this case, the attempt at dismounting +DG1 fails because it is in use by the second database. The restore fails.
- If the second database is offline at the beginning of the restore, Replication Manager dismounts +DG1 and performs a disk restore. In this case, both databases have been restored. This is most likely not what was intended. Furthermore, the second database, not being part of the Replication Manager application set to begin with, wasn't placed in hot backup mode during the replication and therefore may be corrupt.

For these reasons, it is very important to carefully plan the storage layout of ASM diskgroups, based on the restore requirements.

ASMlib volumes revert to their original names during restore

Similarly to ASM diskgroups, Replication Manager reverts the ASMlib volumes to their original names at restore time. The original names are restored along with the restored LUNs after the synch operation is complete. The volumes are always renamed to unique names during a mount operation (see the *ASMlib volumes are renamed during mount* section for details).

Add itional checks

ASM diskgroups can change dynamically very often. Not only can the data extent be rebalanced from one disk to another, but new disks can be added or removed while the diskgroup is still in a mounted state. This has some implications on the restore scenario for Replication Manager.

First, Replication Manager attempts to map the diskgroups and their disk members. It captures, at the time of the replica, a list of the diskgroup names and the disk names that compose the diskgroup.

For example, let's assume that +DG1 is composed of disk1, disk2, and disk3 where these are the logical "label" names given to the disks of an ASM diskgroup. These are not to be confused with the paths to the disks, which are physical paths to the LUNs.

During a restore, if the diskgroup being restored is mounted, Replication Manager executes a query to compare the list of disks for each given diskgroup, as it was at the time of the replica and as it is now on the host. If they don't match, Replication Manager issues a warning. It will not fail the restore, as the fact the disks don't match is not necessarily a fatal condition. It may, however, cause some complications when attempting to bring the diskgroups back to a mounted state, which the administrator must address manually. The warning serves as a reminder to the user that the general layout of the diskgroup(s) being restored doesn't match the original.

If the given diskgroups are not in a mounted state at the time of restore, Replication Manager issues a warning to inform the user that it cannot check the layout and continues with the restore.

Examples of situations in which Replication Manager gives a restore warning:

- If +DG1 was made of disk1, disk2, and disk3 at the time of replication and a disk4 was added after the replication and before the restore, the replica does not contain any information about disk4. Replication Manager dismounts +DG1, and restores disk1, 2, and 3. At this point, disk4 becomes an orphan disk that does not belong to +DG1 anymore as its layout has been restored to what it was at the time of the replica. Disk4 should now be manually excluded by either changing the `asm_diskstring` parameter or changing the permission on the disk4 path. +DG1 can then be mounted and successfully reverted back to its original layout and contents.

- If +DG1 was made of disk1, disk2 ,disk3, and disk4, and disk3 was removed after the replication was taken, then Replication Manager dismounts +DG1 and restores all four disks. Disk3 will be restored as well, as it was part of the replica. After the restore, Replication Manager attempts to remount +DG1. If the `asm_diskstring` parameter does not exclude the path to disk3, the diskgroup mount should succeed. If not, the attempt to mount fails and the user must modify the `asm_diskstring` manually to allow the restored diskgroup to mount.

These simplistic examples are easy to handle because of their size. In production environments, it may not be easy to sort out all the layout changes. For this reason, even though Replication Manager allows the restore in such situations, EMC recommends that administrators discard any replicas taken before a particular layout change and create new replicas, reflecting the current layout of the given diskgroup whenever possible.

ASM diskgroups in RAC environments

RAC environments have the peculiarity of sharing ASM diskgroups across all the nodes of the cluster. For example, +DG1 can be mounted to multiple ASM instances at the same time, which provides the availability of the diskgroup to the database instances of the RAC.

Replication Manager cannot automatically dismount an ASM diskgroup "globally" when running in a RAC-ASM environment. In other words, if we have a three-node RAC database built on a shared diskgroup called +DG1, and Replication Manager is running on node 1, where the restore is occurring, Replication Manager can unmount +DG1 from node 1 but not from node 2 and 3. This has to be done manually.

The restore procedure can be summarized as follows:

1. Manually shut down the RAC database on all nodes.

```
srvctl stop database -d RACDB
```

In case of Replication Manager version 5.4 and later:

Manually shut down the RAC database instance on all nodes except the node on which Replication Manager will run restore:

(In the example, RACDB2 is instance running on node 2, RACDB3 is instance running on node 3)

```
srvctl stop instance -d RACDB -i RACDB2
srvctl stop instance -d RACDB -i RACDB3
```

2. Make the diskgroups inactive.

If the ASM instances running on the RAC nodes only manage the diskgroups involved in the restore, shut down the ASM instances as follows:

```
srvctl stop asm -n node 2
srvctl stop asm -n node 3
```

Note: ASM is not stopped on node 1. In this example, node 1 is where we are performing the restore and we need ASM running there for Replication Manager to proceed.

If the ASM instances running on the nodes manage more diskgroups than what is being restored connect to each individual ASM instance (of node 2 and 3) and manually dismount +DG1:

```
alter diskgroup dg1 dismount;
```

That way, only node 1 has that diskgroup mounted. If +DG1 is mounted in any other node than the one on which the restore is occurring, the restore will fail with a message warning that the diskgroup needs to be dismounted from the other nodes of the RAC.

3. Perform the Replication Manager restore.
4. On node 1, perform the database recovery, then open the database on that node:

   `recover database;` (in its simplest form)

   ```
   alter database open;
   ```

5. Restart ASM on the other nodes and start the RAC database globally:

   ```
   srvctl start asm -n node2
   srvctl start asm -n node3
   srvctl start database -d RACDB
   ```

Support for restore of failover standalone databases in a clustered environment

Replication Manager 5.4 added support for cluster-awareness during restore of standalone databases that are configured to failover from node to node in the cluster. During restore, if the node on which the replica was created is not accessible, Replication Manager runs the restore on another node in the cluster.

Restore considerations with ASM databases

Say a standalone fail-over ASM database is created on shared disk groups, on node1. The other nodes in the cluster are node2 and node3. The following steps need to be performed for restore operation to succeed when initiated from node 1:

1. ASM instances running on other nodes should be shut down manually

   ```
   srvctl stop asm -n node 2
   srvctl stop asm -n node 3
   ```

2. Perform Replication Manager restore.
3. On node 1, perform the database recovery.

   `recover database;` (in its simplest form)

```
alter database open;
```

4. Restart ASM on the other nodes

```
srvctl start asm -n node2
srvctl start asm -n node3
```

Also refer to **Support for RAC awareness during replication of fail-over databases** for pre-requisites. Replication Manager will discover and connect to the RAC ASM instance on the node to which the fail-over has happened and the restore operation will continue to completion.

Restore considerations with non-ASM RAC restores

There is a variety of cluster software on the major platforms. Oracle RAC in its simplest form can be built on Oracle CRS and use ASM as the volume manager. The way to handle a restore in this environment is described above. For specifics pertaining to HP Service Guard, IBM HACMP, SunCluster and VCS using their own version of LVM, refer to the specific procedures described in the *EMC Replication Manager Product Guide* in the "Mount, Restore and Recovery" section.

To perform a restore of a RAC database on these platforms, follow these steps:

1. Manually shut down the RAC database using the srvctl command line utility.
2. Perform platform-specific steps pertaining to the dismount or export of the involved volume groups.
3. Perform the Replication Manager restore.
4. Perform platform-specific steps to reimport the volume groups.
5. Perform the recovery of the database on one node.
6. Restart the RAC database globally using the srvctl command line utility.

Support for RAC awareness during restore of RAC databases

Replication Manager 5.4 added support for cluster awareness during restore of RAC databases. Refer section **Support for RAC awareness during replication of RAC databases** for the configuration pre-requisites.

During a restore operation, if the node on which the replica was created is not accessible, Replication Manager runs the restore on another node in the RAC. At the start of restore, RM now determines the RAC Database instance and ASM instance(for ASM-RAC databases) running on the node instead of retrieving the values of the RAC database instance and ASM instance that were saved during application set creation. So, if a failover has happened after replica creation, the values of RAC database instance and ASM instance(for ASM-RAC databases) would be different now, but by determining the values again dynamically, RM connects to the right instances.

Also, the database init file that was captured during replication would be of a different instance than the one we are currently running on. So, oracle agent renames the init file to reflect the "right" RAC database instance.

Mounting an Oracle replica

Replication Manager 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.

Mounting a Replication Manager Oracle replica

Often Replication Manager replicas are mounted to an alternate host (mount host) to perform some processing on the database without impacting the production database. The processing can range from backing up the replica to tape, running queries for reporting or sanity testing, or other activities facilitated by an offline copy of the database that does not affect the performance of the production database.

The mount options and customer needs are varied and extensive. The following sections detail a sampling of mount scenarios.

Key steps in mounting a replica

Unlike restoring a replica, which allows individual selection of components to restore, the mount functionality is based on an all-or-nothing approach where the replica, as it was taken during replication, is considered as a whole entity.

After the target mount host has been chosen and the mount options have been selected, the Replication Manager client on the mount host:

- Performs the necessary storage technology specific operations to allow the LUNs making up the replica to be visible and ready to use on the mount host
- Imports any volume groups / mounts any file systems
- Optionally performs recovery on the Oracle database and opens it
- Optionally runs any post mount scripts

Specifics regarding production host mount

A special case of the mount functionality is production mount. This capability was added in the Replication Manager product to allow users to mount their replicas back to the production host itself, to an alternate location. Usually this would be used to save the overhead of having an extra mount host or, in certain cases, if a particular file was damaged on the production database (when performing a restore is not an option). In such cases, it is very easy to mount the replica to an alternate path on the production host, and select certain files to copy manually to the damaged production data (surgery).

Because the production host runs the production database, there are restrictions and exceptions associated with this type of mount that restrict the available mount options. Exceptions associated with production mount are described in the following sections.

Oracle objects imported during 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 appear on the mount host in their respective file systems (see the *Alternate path* section for more information), or the corresponding raw devices, raw volumes, or ASM diskgroups.
- **Control file(s)** — If the replica was taken using the consistent split option, the devices containing the current control files are included in their respective raw device(s), volume(s), or ASM diskgroup. These objects will be imported or the file system mounted (see the *Alternate path* section for more information). If the replica was taken using the non-consistent split option, the control file devices are not part of the replica and Replication Manager copies a backup control file to the ERM_TEMP_BASE/‹sid›/ctl/ location instead.
- **Online redo logs** — If Replication Manager uses consistent split to create the replica, the devices containing the current online redo logs are included in their respective raw device(s), volume(s). or ASM diskgroup. Replication Manager imports these objects or mounts the file system (see the *Alternate path* section for more information). If Replication Manager created the replica using the non-consistent split option, the online redo logs devices are not part of the replica and not required for recovery and therefore are not present.
- **Individual archive logs** — If Replication Manager created the replica using the hot backup mode option, the archive logs generated during the hot backup mode period are copied to the mount host in ERM_TEMP_BASE/‹sid›/arch/.
- **Archive log directory** — If you choose to include the archive log directory to the replica, the corresponding file system or ASM diskgroup will be mounted on the mount host (see the *Alternate path* section for more information).
- The directory would be there for structural consistency but does not contain any archive logs, which would be useful for the recovery of the current replica.
- **Initfile/spfile** — Replication Manager moves the initialization file captured during the replication to the ERM_TEMP_BASE location where Replication Manager modifies it to adjust to the various path changes of the key components such as the control files. Replication Manager stores the modified initialization file in the $ORACLE_HOME/dbs/ directory (see also *Using a Custom Initfile during mount* section).
- **Flash Recovery Area** – If the Flash Recovery Area is included in the replica, Replication Manager imports it during the mount operation.

  The following caveats apply to the contents of the replicated FRA:

  - If an alternate path option was used (ASM dg rename or alternate path for file systems), flashback operations will not be possible as the Oracle Database cannot find flashback logs information if the path to the FRA has changed

(Oracle restriction). In that case, the FRA is included as part of the replica but is not usable for flashback operations that target a point in time before the time of the replica. You may choose to restart the flashback function from that point on to start tracking flashback undo changes again.

- If the database rename feature is used, the contents of the flashback logs become invalid and are not usable for flashback operations that target a point in time before the time of the replica.
- If the FRA contains archive logs, be aware that those archive logs do not encompass the full window for the hot backup mode that happened during the replica. That is because the FRA is split at the same time as the datafiles.
- If you are including the FRA in the replica 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 consistent split replication with a mount read write or read only without db rename.

Note: Flashback commands are not integrated into Replication Manager 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.

Oracle mount options

There are three sections on the Mount Options panel of the Job Wizard that relate to mount:

- Generic options (not application-specific) applying to the whole replica
- Generic options applying only to the current component of the application set
- Oracle-specific options related to the data recovery

This section details selected mount options that are related to Oracle and provides guidance to help users decide how to configure the replica.

Alternate path

The alternate path option mounts a file system to a different location from where it was originally located on the production host. This can be achieved using the alternate root or path mapping method. This option only applies to file systems and will have no effect on raw disks, raw volumes, or ASM diskgroups.

This option is used to resolve path conflicts when several replicas 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 types

The least automated mount operation makes all the pieces of the replica available on the mount host, ready for use. In many cases, that is all that is required of Replication Manager. Once that is complete, a third-party backup utility can apply its own processing to complete the backup.

Other mount scenarios require Replication Manager to perform an automatic restart of the Oracle database. Different scenarios require this procedure to occur in different ways and Replication Manager can accommodate different restart scenarios.

The Replication Manager Oracle agent takes into account many parameters when performing a restart operation on the database, including:

- **Type of replica** — Consistent split versus non-consistent split; hot backup mode versus non-hot backup mode
- **Alternate root or path mapping** — Makes allowances for changes to the path used to access different pieces of the database (also covering the raw disks and raw volume name changes that occur during the mount)

The following sections describe the various types of mounts that are available:

File system mount only

This is the simplest form of mount, achieved by selecting the "Do not perform database operations" radio button. Replication Manager does not automatically initiate a recovery or restart operation. This mount option completes after the LUNs have been made visible to the host, volume groups have been imported, and the file systems have been mounted. All other processing is completed outside of Replication Manager.

Note: Typically, Replication Manager tracks changes to the replica during a mount (such as recovery of the data for example), to ensure that the same replica can be unmounted and successfully remounted at a later time. If the replica is mounted using this mode, Replication Manager will not track changes to the database. Therefore any manual changes made to the database outside Replication Manager may compromise the replica and prevent a remount later (depending on the nature of the change and the recovery option selected on the subsequent mount).

This option is mostly used in cases where the database does not need to be recovered/started. This type of mount preserves the original unaltered state of the replica. The advantage of this option is that it does not require Oracle to be installed on the mount host, except on Windows systems where the Oracle binaries do need to be installed regardless.

Prepare only mount (before version 5.2.2) and "Generate scripts for manual recovery" (5.2.2 and later)

The prepare only 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.

The prepare only mounts perform the following steps automatically:

- Performs the initialization file modifications necessary to adjust to the various path changes affecting components of the Oracle database
- For Oracle releases up to and including 11g R1, creates an ASM initialization file if ASM is involved with the asm_diskstring parameter updated to contain the raw disks making up the current replica.
- For Oracle 11g R2, if ASM is involved, Replication Manager generates an asm_steps.txt file in the /ERM_TEMP_BASE/‹SID_NAME›/ location (where SID_NAME is the database SID) on the mount host. This file gives a list of manual steps to be performed on the mount host's ASM instance before database recovery can be done.

  Note: If the *ASM diskgroup rename* option is chosen, the ASM diskgroups contained in the replica are renamed automatically, even though neither the ASM instance nor the DB instance will be started.

- Generates scripts, as follows:
  - db_rename_save_script.sql — Script for mounts that specified the db rename option (re-creation of the control file).
  - normal_recovery_save_script.sql — Script for mounts that simply recover the database without renaming it.
  - rman_recovery_script.sql — RMAN script created to perform the actual recovery of the database

The same note regarding manual changes in the "File system mount only" section applies here as well.

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 replica but also have some expanded recovery options available if needed.

Read only mount (not available if the user who created the replica is a non-SYSDBA user)

The read only mount performs the following steps automatically:

- Starts the Oracle instance and brings it to a mounted state (control file is read). If the replica was taken using the consistent split option, the control files from the mounted replicated LUNs are used. Otherwise, the control file imported in ‹ERM_TEMP_BASE/‹sid›/ctl/ is used.

- Informs Oracle of the following changes to the environment:
  - Path changes if an alternate path option is being used
  - Raw disk changes
  - Third-party raw volume name changes
  - ASM diskgroup name changes

It does so by running an internal SQL*Plus script that runs standard Oracle commands such as:

```
alter database rename file 'xxxxx' to 'yyyyy'  ;
```

- Recovers the database using SQL*Plus commands that depend upon the replica type. Replication Manager 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). Replication Manager presents a mounted database matching the point in time when the replica was created.
- Opens the database in read-only mode, thus not allowing any more changes than what was 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 replica itself *is* altered however, in that it has been recovered and logs have been applied. However, the actual data contained in the database is not changed.

Note: If the "rename database" option is used in conjunction with the "Read Only" option, the database changes significantly. The newly renamed database would consequently be opened in read only mode, so as not to allow any data change, however, the significant changes to the database may have implications for future mount or restore operations.

Read write mount (not available if the user who created the replica is a non-SYSDBA user)

The "Read/Write" mode is almost identical to the "Read Only" mode because Replication Manager performs the same steps to bring up the instance and update the paths to the relevant components. The major difference relates to how Replication Manager opens the database at the end of the mount operation. As the name suggests, Replication Manager opens the database in "Read/Write" mode, thus allowing further changes to the database once it is mounted.

Depending on whether the replica was created with or without consistent split, there is an additional key difference that comes into play as well.

- **Consistent split replicas** — Since the online redo logs are available, the database opens normally once the recovery completes.
- **Non-consistent split replicas** — These replicas are exclusively dependent on the archive logs and backup control file to recover the database, therefore this type of replica is considered a point-in-time recovery (also called incomplete recovery). For that reason, Oracle forces Replication Manager to open using the "resetlogs" clause, which effectively creates a new incarnation of the database, restarting the archiving as sequence #1.

- When opened with the resetlogs, the database is no longer the same database it was before the mount (even though the content has not changed). Such a replica, for example, could not be restored to production later, because the archive logs from the production side would no longer be compatible with the altered copy of the database.

In general, replicas that have been mounted in "Read/Write" mode should not be restored to the production database, whether the resetlogs clause was used or not. Whenever a database is mounted in "Read/Write" mode, 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 and the data is used for testing and manipulations.

The Oracle password file is also re-created (see the *Assign new sys password* section for more details) in both the "Read Only" and "Read/Write" modes, using the Oracle orapwd command line.

Catalog with RMAN (not available if the user who created the replica is a non-SYSDBA user)

See the *Integration with Oracle Recovery Manager (RMAN)* for more details.

Celerra NFS replicas and Oracle mount options

The Celerra NFS snapshots can be mounted in either a Read/Write or Read Only mode, from the point of view of the filesystems. When the snapshot is being mounted as read only, Oracle will not be able to write to the datafiles to perform recovery (which is required to bring the database to an open state (Oracle ReadWrite or ReadOnly state). Hence, the "recover and open" option has been disabled if the NFS option for the selected snapshot is Read Only.

Database rename

The database rename feature allows the user to mount the database after it has been renamed. The instance starts and the following tasks occur automatically:

- Replication Manager re-creates the control file(s) by running an internally generated SQL*Plus script, which uses the create controlfile command to reset the name of the database.
- Once the control file has been re-created with the new database name, Oracle modifies the datafiles and online redo logs to apply the new database name and database ID.
- Replication Manager recovers the database as described earlier.
- Replication Manager recovers the database and opens it in either "Read Only" mode or "Read/Write" mode (" Read/Write" mode imposes mandatory resetlogs). This choice depends upon the recovery type.

Note: If the recovery type was "Prepare only," the database is not actually renamed or recovered. Rather, a script containing the generated "create controlfile" command is created in ERM_TEMP_BASE/‹sid›/ and can be used as a stub template to rename the database manually.

Renaming the database is a very intrusive option that forever alters the composition of the database. At that point, it becomes incompatible with the original database on the production host. For that reason, a replica that was mounted with the database rename option is not suitable for restore. This option is exclusively used in repurposing where a production environment is cloned and Replication Manager creates a new incarnation of the database on the mount host.

Another reason to rename the database is when you are mounting a replica to the production host. In order to start the mounted replica on the production host, you must use an alternate path and rename the database, so that the two copies of the database (production and replica) can coexist side by side on the production host.

SID rename

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 replica in any way.

Note: Replication Manager creates directories based on the SID name when performing mounts. The use of the SID rename can sometimes be leveraged to allow several replicas 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. Let's say we have a database called PROD with a corresponding SID called PROD. To mount A replica of PROD to the production host, on an alternate path, without actually starting it, select the Prepare Only option and choose "SID rename." Rename the SID to PROD2 (for example). This will inform Replication Manager that the database copy on the replica must not be started and the alternate SID name PROD2 is provided so that directory structures collisions do not clash with that of PROD's.

To actually start PROD2 on the production host, select the Database Rename option as well, to prevent a database clash with PROD at startup time.

Assign new sys password

As part of the mount operation and creation of the Oracle instance on the mount host, Oracle requires a password file to be re-created using the Oracle orapwd command. As part of that procedure, a new password is assigned for the built-in SYS user.

The password supplied here does not need to match the SYS password of the original database. It is the new password for the SYS user on the new instance of the database running on the mount host.

Note: The password file keeps a list of the users who have been granted SYSDBA and SYSOPER privileges. If the recovery type is "Read Only" or "Read/Write", Replication Manager regrants SYSDBA and SYSOPER privileges to the same users who had such a privilege on the production database. So if four users have SYSDBA or SYSOPER privileges (determined by querying the v$pwdfile_users view on the production database) those same users are granted SYSDBA or SYSOPER privileges on the mounted copy.

Oracle home

This is more of a parameter rather than an option. This specifies the ORACLE_HOME path to use for the mount. Because there is no reliable way for Replication Manager to validate the accuracy of the selected ORACLE_HOME to be used for the mount host, the user should verify the value for this parameter.

Replication Manager 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 (\database\ on Windows)
- 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.

Fail if the SID exists

In an attempt to avoid overwriting an existing instance on a particular mount host, Replication Manager runs three checks to detect the presence of a particular SID on the system. These checks are as follows:

- Check $ORACLE_HOME/dbs/ for an init‹sid›.ora file that matches the SID that is about to be mounted. If the alternate SID option is selected, the search is based on the new SID name. For example, if the production SID is PROD and the alternate SID name chosen for the mount is PROD2, Replication Manager verifies that the following filename does not exist: $ORACLE_HOME/dbs/initPROD2.ora

- Check for $ORACLE_HOME/admin/<sid>/ dump directories and verify that they do not exist.
- Check for an indication that the target SID has been mounted in the local oratab file of the mount host or, on a Windows system, in the registry.

If any of these checks reveal evidence that the SID exists on the mount host, the mount fails. The checks are not designed to be exhaustive. For instance, the Replication Manager Oracle agent does not actually check to see if the corresponding Oracle Instance is running. But the existing safeguards are typically sufficient to avoid collisions.

These three checks can be overridden in Replication Manager 5.0 SP2 and later releases to reduce false positives that cause too many mount failures. Users can choose to skip these checks and effectively continue the mount regardless of the potential presence of the SID on the hosts. This could save considerable time in scenarios where:

- Environments are large.
- Environments contain a lot of LUNs.
- Excessive failures become a problem.

Caution is advised when using this option when the target is the production host, as this may result in overwriting production files.

Operating system user for mount

This feature is only available on UNIX-based versions of Replication Manager.

The operating system username feature is the mount host counterpart of those credentials provided during the application set creation. Ideally, the username provided there should be the operating system user, owner of the Oracle binaries installed on the mount host. Therefore, this field defaults to “use userID of Oracle binaries.”

Just as during the replication and restore operations, Replication Manager spawns a child process from the Replication Manager client's main process, and assumes the identity of the provided operating system username. That child process executes Oracle commands and queries. This ensures a safe and stable execution environment from which to run the Oracle functions.

An operating system user different from the one owning the Oracle binaries on the production host may be supplied. The user must exist on the mount host but does not have to match the user of the production system, as was required in Replication Manager versions prior to 5.1.

In cases where ownership differs, the Replication Manager Oracle agent automatically performs “chown” statements on the required pieces of the database, such as datafiles, control files, redo logs, and so on.

Note: If the ownership of the files contained in the replica was changed as part of the mount, they remain changed on the replica even after it has been unmounted. However, if the replica is restored later on, Replication Manager restores the original ownership of the files.

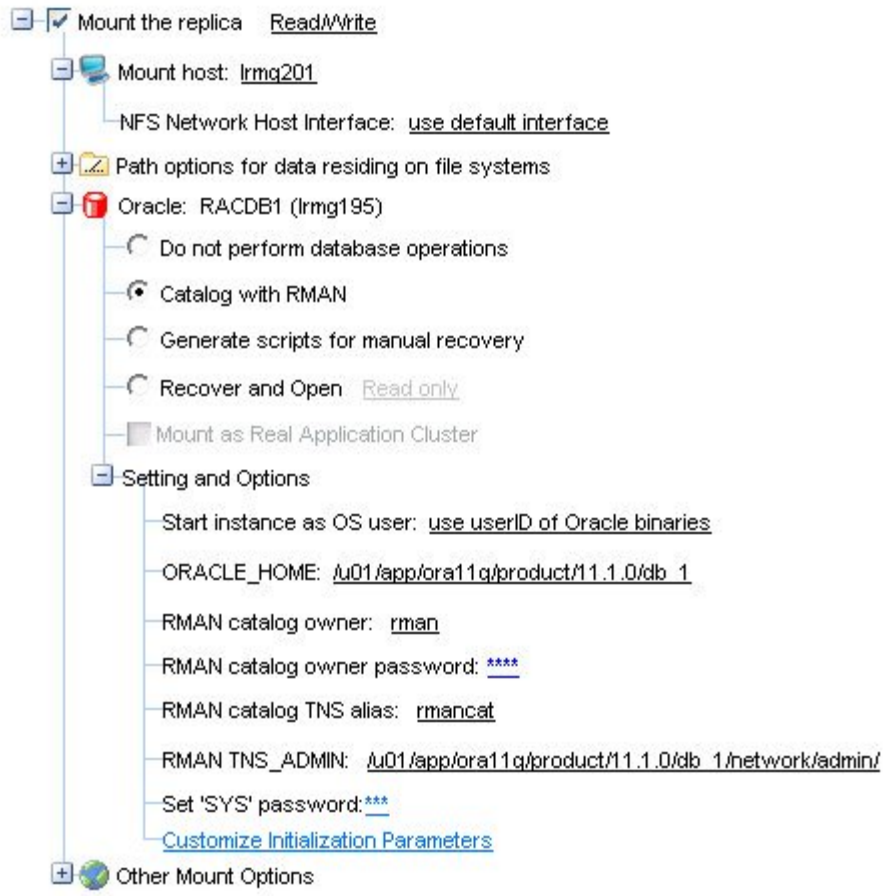

Figure 11. Mount options

The mount options panel has been reorganized in Replication Manager 5.2.2. The main "recovery modes" have been renamed and laid out as three radio buttons. The following is a mapping of the pre-5.2.2 options and 5.2.2 options:

Table 2. Mapping of pre-5.2.2 to 5.2.2 options

Pre Replication Manager 5.2.2	Post Replication Manager 5.2.2
Clear the "Recover the database" checkbox	Select the "Do not perform database operations" radio button
Check the "Recover the database" checkbox Set the "Recovery type" to "prepare only"	Select the "Generate scripts for manual recovery" radio button
Check the "Recover the database" checkbox Set the "Recovery type" to "Read only"	Select the "Recover and Open Read Only" radio button
Check the "Recover the database" checkbox Set the "Recovery type" to "Read write"	Select the "Recover and Open Read Write" radio button

The contents of this paper have evolved from the original Replication Manager version for which this paper was written and the old and new naming conventions are used interchangeably across this document.

Customizing the initialization parameters used for mount

Use the Replication Manager 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 production initialization file used with the mounted database. In previous versions of Replication Manager, customized parameters were entered by editing a special init file. Any existing customizations in this file will continue to be recognized but use of the console is encouraged.

When the mount operation runs, this init file will be combined with the existing init file to create a new version of the init file used with the mounted database.

Setting parameters with the Replication Manager Console

To customize initialization parameters, click Customizing Initialization Parameters under Mount options (in the Job wizard, Job properties, or Mount wizard). The Customize Initialization Parameters window appears.

The following operations are available:

- To add a new parameter, click New and enter a parameter name and value.
- To disable a parameter but not remove it from the list, clear the checkbox.
- To remove a parameter, select it and click Delete.

Notes on setting custom initialization parameters

When setting initialization parameters using the Replication Manager Console:

- The values that you set here override any parameters that are set in an init file.
- Only the parameters that were added using this table are displayed in the list.
- Use care when specifying parameter names and values. An invalid entry can have an undesirable result (for example, it can prevent the Oracle instance from starting).
- No special permissions are required of the user account running Replication Manager Console.
- The following parameters are not settable in the Replication Manager Console:
  - control_files
  - db_recovery_file_dest
  - log_archive_dest*
  - db_name
  - instance_name
- In general, if the file or directory specified by a parameter value does not exist, Replication Manager does not create it. The exceptions are values for background_dump_dest core_dump_dest and user_dump_dest. In these cases, Replication Manager creates the directory on mount and deletes it on unmount.
- In 11gR1 and later, the dump directories parameters still exist but are deprecated in favor of diagnostic_dest. In that case, if you set any of the *_dump_dest parameters to a value, they will be ignored in favor of an automatically generated directory under the location pointed to by diagnostic_dest.

Unmounting Oracle replicas

Unmounting an Oracle replica is the opposite of mounting a replica. When you perform an unmount, Replication Manager completes the following actions:

- Shuts down the running database if it had been started by Replication Manager. (If the "Prepare Only" or "Filesystem Only" modes were used, Replication Manager does not attempt to shut down the database; that is the user's responsibility.)
- Shuts down the created ASM instance, if ASM is part of the environment. Refer to the section *ASM model with Replication Manager during mounts* for details.
- Unmounts file systems, if file systems are part of the environment.
- Deports/exports third-party volume groups.
- Updates the Replication Manager server with the necessary information required for Replication Manager 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. Therefore, "Prepare Only" and "Filesystem Only" mounts display warning messages explaining that Replication Manager cannot guarantee the success of future mounts of the replica.

Note: Replication Manager does not maintain the "chain of events" on replicas mounted using "manual" modes ("Prepare Only" or "Filesystem Only"). In other words, if the replica was previously mounted using a manual mode and modified significantly (recovery, path changes), Replication Manager may not be able to remount that replica later on with an automatic mode such as "Read Only" or "Read/Write".

As described in the section *ASM instance user privileges*, for ASM databases created in Oracle 11gR2, an ASM instance user with SYSASM privileges is required to perform tasks in Replication Manager. The ASM instance user info provided is used during mount to perform the ASM asm_diskstring and asm_diskgroup operations during mount and unmount. In cases where SYSASM privileges are revoked from the ASM instance user before the unmount operation, the operation fails.

Integration with Oracle Recovery Manager (RMAN)

Starting with Replication Manager version 5.2.3, Replication Manager offers integration with Oracle Recovery Manager (RMAN) in environments with Linux operating systems and Celerra NFS storage.

Starting with Replication Manager 5.3, RMAN support has been extended to Solaris platforms for databases residing on ASM, NFS and regular filesystems. Replication Manager 5.3.1 has further expended it to HP platforms (ASM, NFS and filesystems as well).

RMAN catalogs instances of Oracle that exist in a customer's environment and can manage the backup and recovery of Oracle instances. With the latest version of Replication Manager, this integration allows Replication Manager to facilitate the cataloging of mounted replicas using RMAN. This section describes this functionality.

Oracle Recovery Manager prerequisites

There are specific prerequisites that must be met in order for Replication Manager to integrate with RMAN. These prerequisites are described here:

- RMAN catalog database must exist and be accessible on the same network as the mount host.
- The tnsnames.ora file on the mount host must contain a tnsalias that points to the RMAN catalog database where Replication Manager should catalog the replica.
- The catalog and catalog owner must be created prior to mounting a replica to be cataloged.
- Production database must be registered in the RMAN catalog before mounting the replica.
- 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.
- Production databases that integrate with RMAN must be at least Oracle 10g.

Mounting with the "Catalog with RMAN" (Recovery Manager) option

Choosing the option **Catalog with RMAN** performs the following operations to facilitate the use of RMAN with Replication Manager replicas:

- Starts the Oracle instance on the mount host and brings the database to a mounted state
- Database remains unopened
- Replication Manager automatically catalogs the components of the mounted Oracle replica in the RMAN recovery catalog using the following RMAN commands:
  - catalog datafilecopy
  - catalog controlfilecopy. Note that Replication Manager creates a copy of the backup control file before starting the database instance. That is so that Replication Manager can present an unaltered copy of the backup control file for cataloging purposes (the backup control file that is used to start the database instance get modified when updating the path to the datafiles, for example).
  - catalog archivelog
- Replication Manager automatically uncatalogs the same components of the Oracle replica from the RMAN recovery catalog when the replica is unmounted.

Components that are cataloged include:

- Datafiles
- Backup control files

- Archive logs
- Flash Recovery Area contents (if enabled)

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

- View the contents of the cataloged replica(s) using RMAN commands such as: list datafilecopy all;
- When specifically mounting to the production host, RMAN cataloging of the replica's contents allows the following operations to be possible from the RMAN command line utility:
  - Perform RMAN individual file restore from the mounted replica. 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 replica. 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 replica. 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 Replication Manager replica stays mounted/accessible to the production host.

Using the BCT file with RMAN incremental backups

As explained in the *Block Change Tracking file (BCT file)* section, the BCT file can improve the 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 the Replication Manager replica which just got mounted to a host with the **Catalog with RMAN** option.

While Replication Manager does not get involved with actually running any RMAN backups, it aims to facilitate this. Along with bringing the database to a mounted state and adjusting the datafile paths before cataloging with RMAN, Replication Manager 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 couple examples below show simple versions of RMAN backup commands which would, in this case, leverage the BCT file, since it was selected to be included during the replication.

RMAN backup command against the first replica:
(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 replica.  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, you can 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 replica cataloged with RMAN

Whenever Replication Manager unmounts an Oracle replica mounted using this option, the replica is uncataloged from RMAN to indicate that it is no longer available for the recovery operations.

Notes and restrictions

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

- Replicas created without hot backup mode are not eligible for integration with RMAN.
- Replicas mounted with RMAN integration cannot be renamed using Replication Manager’s database rename option.
- Read-only replicas cannot be cataloged using RMAN.
- Replicas of databases built on ASM storage cannot be mounted to the production host in conjunction with the **catalog with RMAN** option (except for 11gR2 databases where this is allowed; note that further releases of Replication Manager will remove this restriction all together)

- Replication Manager can be configured to skip the cataloging of Oracle data files. This can be accomplished by setting the following environment variable to 1:

```
ERM_ORACLE_RMAN_NO_CATALOG_DATAFILES
```

If you do not set the environment variable, all datafiles and logs will be cataloged. In some cases it is desirable to prevent the cataloging of datafiles, since that is not required for backups and skipping that cataloging can improve backup performance in environments with many datafiles. To skip the cataloging of datafiles, set the following environment variable:

```
ERM_ORACLE_RMAN_NO_CATALOG_DATAFILES=1
```

You can modify the rc.irclient script to set the environment variable. You must restart the irccd daemon for the setting to take effect.

ASM model with Replication Manager during mounts

As previously explained in the ASM section, the Replication Manager Oracle agent needs to interact with the ASM instance in order to discover, mount, and dismount ASM diskgroups.

On the production system, during a replication or restore operation, Replication Manager interacts with the ASM instance serving the database and gathers the appropriate ASM instance name via the Application Set Wizard (usually +ASM or +ASM‹node number› in RAC environments).

Behavior for Oracle versions through Oracle 11gR1

During the mount operation, the Oracle agent does not use any existing ASM instance that may be running on the target mount host. Instead, Replication Manager generates its own ASM instance and customizes the ASM instance parameters like asm_diskgroup and asm_diskstring to meet the needs of the current mount.

Replication Manager generates a random unique name for the ASM instance, for example +RM‹unique_identifier›. Replication Manager generates one ASM instance per mounted replica. This model was implemented so that each replica can maintain its independence, and not interfere with other replicas, especially since several copies of the same diskgroup may be present on the target host.

Note: Replication Manager shuts down and removes the temporary ASM instance upon replica unmount.

On mounts using the “Prepare Only” option, the ASM instance that Replication Manager generates on the mount host will not be started. However the corresponding init‹+RM‹number›.ora file will be generated in $ORACLE_HOME/dbs/ so that the customer can perform a manual start of the ASM instance if desired.

Even though Replication Manager takes care of generating and starting its own ASM instance, it assumes that Cluster Manager software is running on the target host. The “cssd” daemon is one of the key components that allow ASM to function. When the

Oracle software is installed and ASM is configured for use on the host, the Oracle installer prompts the user to run the "$ORACLE_HOME/bin/localconfig reset" command in order to enable and start the cssd daemon. This only needs to be done once, and must occur before Replication Manager can successfully start ASM instances on the target host.

Behavior for Oracle version 11gR2

In Oracle 11gR2, multiple ASM instances on a host are no longer supported. Therefore, the earlier model that generates an ASM instance for each mounted replica no longer applies.

1. For Oracle 11gR2, during the mount/unmount operation, the Oracle agent connects to the existing ASM instance running on the mount host to manipulate the asm_diskstring parameter to mask/unmask devices during mount/unmount of Replication Manager replicas. This requires the ASM instance on the mount host to be started before a mount operation is performed. Also, the ASM instance on the mount host must have a valid and accessible "spfile." This is a mandatory requirement to make the asm_diskstring and asm_diskgroup operations on the ASM instance persistent.
2. There are three new options in the mount options panel of the Replication Manager Console as follows:
   - ASM instance name
   - ASM username
   - Password (of the ASM user)

Note: The ASM username entered in the mount options panel must have SYSASM privileges. It can be SYS or any other user, as long as it has SYSASM privileges.

Figure 13. ASM options in the mount panel

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 asm_username/password 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 replica was mounted and the ORCL:* disks are those of the replica being mounted.

```
alter system set asm_diskstring = 'Path1','Path2',
'ORCL:EMCERM84140742700','ORCL:EMCERM84140742701','ORCL:EMCERM8414074270
2','ORCL:EMCERM84140742703' 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 replica was mounted and DG1, DG2 are the diskgroups of the replica.

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

6. Mount the diskgroups in the replica using the following command:

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

Replication Manager automatically performs these steps during mount in Read-only and Read-write operations. For the mount option “Generate scripts for manual recovery”, the steps above must be performed by the user manually if the database will be recovered on the mount host.

Similarly, during unmount, the 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 replica 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 replica. In the example below, it is assumed that DiskGroup1 and DiskGroup2 were the diskgroups present on the mount host before the current replica 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 replica. In the example below, it is assumed that Path1 and Path2 were the diskstrings present on the mount host before the current replica was mounted.

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

Replication Manager automatically performs these steps during unmount if the replica was mounted in Read-only or Read-write recovery mode.

For the mount option "Generate scripts for manual recovery," the steps above must be performed by the user manually. The steps are listed in the unmount section of the asm_steps.txt file that was generated by Replication Manager in the /ERM_TEMP_BASE/<SID_NAME>/ location during the mount operation (where SID_NAME is the database SID) on the mount host.

ASM diskgroup rename (UNIX and Linux only)

Replication Manager 5.1 introduced a new feature that allows you to rename ASM diskgroups within a replica. The advantage of renaming ASM diskgroups is that it facilitates the following mount scenarios:

- Production mounts of replicas containing ASM diskgroups
- Multiple concurrent mounts of replicas containing copies of the same diskgroups

When an ASM instance handles a particular ASM diskgroup, it reads the headers of disks it has visibility to and determines the appropriate diskgroup for each disk. When all disks of a given diskgroup are found and no duplicate is found, the diskgroup can be mounted. If more than one copy of a particular disk is found, ASM issues errors about duplicates. Also when a diskgroup is mounted, a second diskgroup with the same name cannot be mounted. Refer to Figure 12 and Figure 13 for illustrations of the issues.

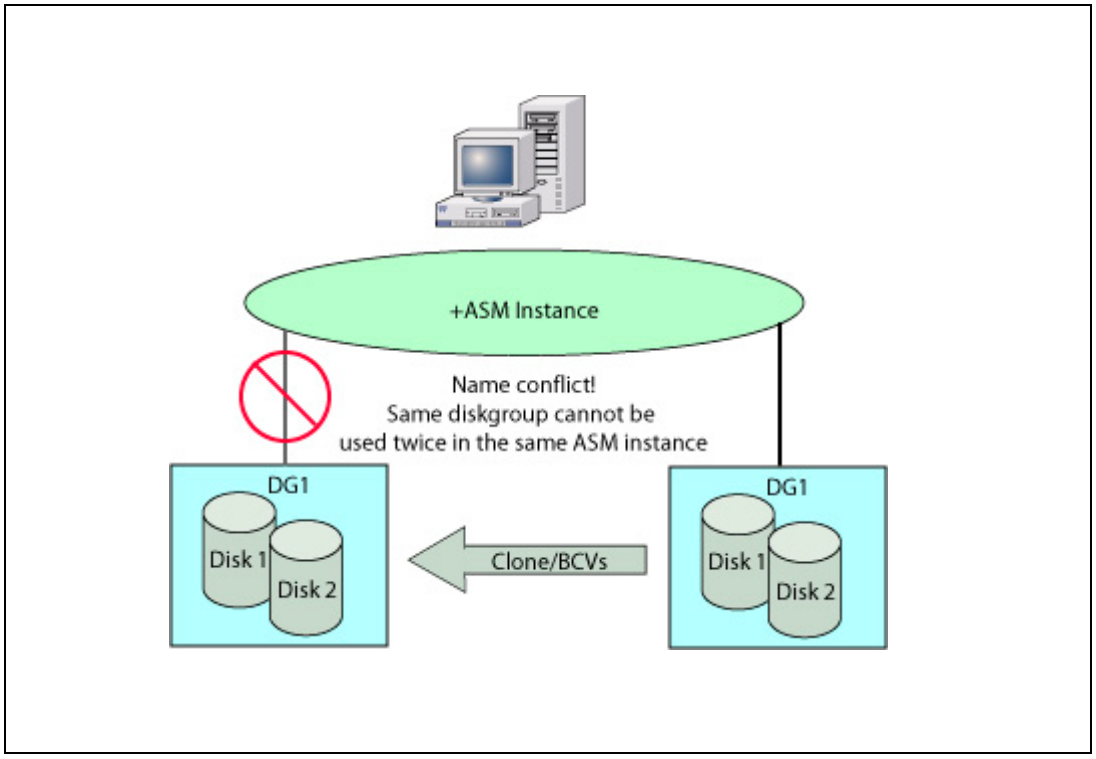

Figure 12. Name conflicts when cloning an ASM diskgroup to production

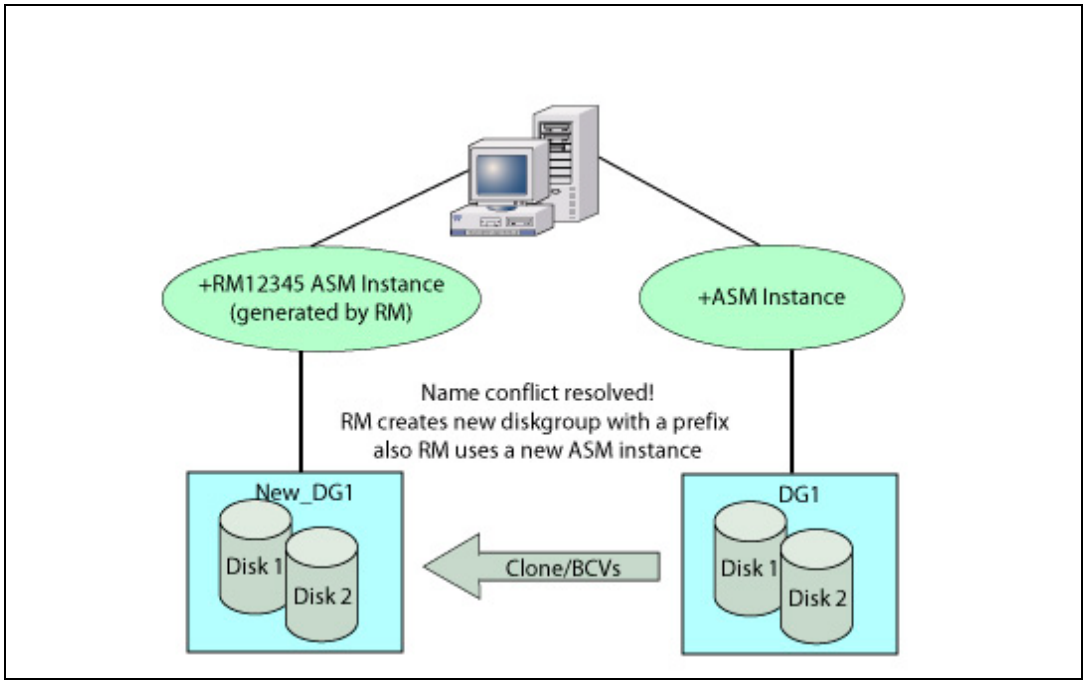

Figure 13. Name conflicts resolved by Replication Manager

The Oracle 10*g* and 11*g* R1 kernels do not possess a rename function for ASM diskgroups. A Perl script was, however, developed by Oracle as an interim solution to allow overwriting the disk headers using an Oracle executable "kfed" underneath the covers. This executable is not built by default when the Oracle Database software is

installed; however, it can be built using the provided makefile in $ORACLE_HOME\rdbms\lib\. Building that executable is a required step if you intend to use the ASM diskgroup rename mount option or if you are running replication jobs to target devices that are visible to the production host.

In the case of Oracle11gR2, the executable kfed is pre-built and does not have to be manually built. In case of Oracle 11gR2 on Linux, instead of kfed, Replication Manager uses renamedg, a tool for renaming ASM disk groups. This tool, provided by Oracle, requires patch 9316059 to be installed as a pre-requisite for diskgroup rename functionality. Please refer Readme.txt provided with 9316059 for details on how to apply the patch. The renamedg tool is also supported on Solaris from Replication Manager version 5.3.2 onwards.

To specify a new name for the diskgroups that are being mounted as part of the current replica, use a prefix supplied in the mount option panel. Replication Manager prepends this prefix to all the diskgroups contained in the replica. For example, if the original production diskgroups were called DG1, DG2, and DG3, and the prefix chosen was New_, Replication Manager would rename them to New_DG1, New_DG2, and New_DG3. Also, for "Read Only" and "Read/Write" mount options, Replication Manager would handle the updated paths of the various objects residing on the diskgroups, that is, +DG1/dbname/datafiles/myfile1.dbf would become +New_DG1/dbname/datafiles/myfile1.dbf

ASMlib volumes are renamed during mount (Linux platforms only)

Just like ASM diskgroups, the ASMlib volumes on which they are built need to have unique names on the host. Failure to create unique names confuses the ASMlib driver and can cause it to choose the wrong version of an ASMlib volume visible to the host. For this reason, Replication Manager always renames ASMlib volumes when mounting a replica to a host to ensure that the volume names are unique. This ensures that multiple replicas visible to the same host properly identify their version of a given ASMlib volume.

Unlike the ASM diskgroup rename option, the new names of the ASMlib volumes are not configurable at the user level given the extreme importance of keeping them unique on the host and the fact that they do not affect the database object paths like the ASM diskgroup names do. Replication Manager automatically generates unique names for ASMlib volumes. The volume names would look like this example:

EMCERM 909269955 00.

Production host ASMlib volumes clobbering

The uniqueness of the ASMlib volume names is even more important when mounting the replica to the production host because it is guaranteed that ASMlib volumes will have the same name as those included in the replica.

This presents a problem as soon as the clone LUNs are visible to the production host. In some cases, the clone LUNs become visible immediately. That is the case with CLARiiON replications on Linux where the clone LUNs must be surfaced ahead of time

to the target host. In this situation, a conflicting copy version of the production ASMlib volume would immediately become visible to the production host as soon as the replication is complete. This in itself would not disturb the ASMlib layer as long as the host stayed booted. Upon reboot however, there would be no guarantee of the order in which the ASMlib driver would choose to pick its devices and so potentially could pick the wrong version of the ASMlib volume.

To compensate for this, Replication Manager modifies the clone LUNs immediately after the source and clone LUNs are synced together. The clones get their ASMlib labels renamed to a unique number immediately to avoid confusion with the production LUNs. This ensures that even a reboot of the host would not confuse the ASMlib layer, which is serving the ASM diskgroups on the host.

Support for a separate ORACLE_HOME for ASM

Replication Manager version 5.2 and later support the use of two separate ORACLE_HOME directories, one for ASM instances and one for Oracle databases. Refer to the *ASM connection and authentication* section for more specific information.

Support for mounting a RAC ASM replica to a target RAC

Replication Manager 5.2.2 added the capability to mount a replica taken from a Real Application Cluster (RAC) database on Linux to another RAC (or to the production RAC itself) and preserve the clustered nature of the database. Prior to that release, Replication Manager could only perform a standalone mount.

Note: Starting from Replication Manager version 5.3.1, mounting from one ASM-RAC environments to another is supported in Oracle 11gR2.

During a standalone mount, Replication Manager processes the initialization file during the mount operation. In Replication Manager releases prior to version 5.2.2, all parameters related to cluster operations (such as, cluster_database) are set to OFF. With Replication Manager version 5.2.2 on Linux platforms running RAC ASM, there is a new option on the Oracle mount options panel called **Mount as Real Application Cluster** that offers the option to mount to a RAC cluster.

The initial design for mounting cluster-based replicas as standalone was perfectly suitable for backups. Regardless of how many nodes were running the production clustered database, only one node or host is needed to perform a backup. However, a growing demand for repurposing support and performance testing on a non-production copy of the database has led to this new cluster mount functionality. The configuration of the cluster to which the replica is mounted must match the production environment, typically with the same number of nodes on the production and mount clusters.

The purpose of this new feature in Replication Manager is not to configure a cluster, add nodes or hosts to the cluster, or perform push installs, OCR, and voting disks configurations. Rather, the above are prerequisites to using this feature. In other words, the target cluster must be created, configured, and operational before mounting the replica with the **Mount as Real Application Cluster** option.

This means that the following are prerequisites for Replication Manager:

- ASM must be the LVM that stores the database files
- ASMlib must be used to ensure that all volumes that compose the various ASM diskgroups appear with the same names on all nodes of the cluster
- SSH must be configured as well as user equivalence for the Oracle operating system user and root. This is to allow propagation of initfiles across the different nodes of the cluster. In case of Oracle 11gR2, in addition, SSH needs to be configured for the grid user as well.
- Oracle Clusterware and Oracle database software must be installed and configured with the same paths and operating system owners on all nodes.
- The Replication Manager client does *not* need to be installed on all nodes of the cluster. Only the mount host.

The process that Replication Manager follows to achieve Real Application Cluster mount of a replica is:

- A standalone mount is performed at first on the mount host (the mount host needs to be part of the target cluster). Regardless of how many nodes exist in the target cluster, only one host is selected for the mount of the replica: the traditional mount host. This step is done so that the database is recovered and ready to be opened.
- For Oracle versions through 11gR1, an ASM instance name is generated by Replication Manager and started as part of the standalone mount. This provides the diskgroups for the database instance that accesses the data. For Oracle 11gR2, Replication Manager connects to the ASM instance running on the mount host and sets the asm_diskstring and asm_diskgroups parameter on the mount host ASM instance. It then mounts the diskgroups on the mount host using the sqlplus command:

```
alter diskgroup <DG_NAME> mount;
```

- Replication Manager shuts down the database once the recovery is complete.
- Replication Manager propagates the database instances to the various nodes of the cluster.
- Replication Manager generates the database instance names for each node based on the following algorithm: The mount host gets the first instance (“1”), the other nodes get the incremented instance name. For example, if the original database was RACDB, the mount host will end up running RACDB1, the following node will run RACDB2, and the potential third node will run RACDB3.
- A “master” database initialization file is created, containing all parameters for all instances of the cluster.
- For Oracle versions through 11gR1, an individual initialization file is created for each ASM instance. For Oracle 11gR2, Replication Manager sets the

asm_diskstring and asm_diskgroups parameters for all other ASM instances running in the cluster. It then, mounts the replica's diskgroups on all nodes(other than the mount host) of the cluster using the srvctl command:

```
srvctl start diskgroup –g <DG_NAME>
```

- The initfiles are copied over the remote nodes using scp, once the initfiles are ready.
- Once all the pieces are in place, remote invocations of Oracle's tool *srvctl* are performed for each ASM instance and database instance (srvctl add …).

Note: For Oracle 11gR2, this step needs to be done only for the database instance.

- The ASM instances(only for Oracle versions prior to 11gR2) and database instances are registered in the target cluster and ready to be started, which is done by a call to "srvctl start database …"

In order to be able to use the "Mount as Real Application Cluster" feature successfully, there are a few Replication Manager functional requirements to observe:

- The archive logs/FRA devices (which are usually optional to include) must be included in the replica. The reason is that Replication Manager leverages the clustered nature of the ASM diskgroups and thus their visibility across the cluster.
- The consistent split replication option must be used. The reason is that, in this mode, the control files and online redo logs devices get automatically included in the replica, which is required so that all target nodes consuming the replica can see those devices.
- The "CRS home" (where Oracle Clusterware is installed on the mount host) must be entered in the Oracle mount option panel.
- Replication Manager does not allow change to the SID name when **Mount as Real Application Cluster** is selected.
- Replication Manager does not require or use the listener on the mount host and remote nodes, nor does it make attempts to configure the listener.ora or tnsnames.ora files. If listening services are required, they need to be configured by the user ahead of time.
- Other traditional Replication Manager Oracle mount options are fully supported with this feature: database rename, ASM diskgroup rename, production mount to the original cluster.
- The dual homes feature, which allows the ASM software to be running out of a separate home than the database software, is supported with this feature as well, with the restriction that the operating system owner of the two sets of binaries needs to be the same, as well as the group.
- In case of Oracle 11gR2 RAC configurations, all the nodes of the target cluster should have a valid and accessible spfile. Also, the ASM instances on the target

RAC to be started before a mount in recovery are initiated using Replication Manager.

- **Mount as Real Application Cluster** for Oracle 11g R2 RAC One Node database replicas is not supported.

Impacts of mounted replicas on restore

Because the mount options have different purposes and goals, and some of them modify the replica, it may be difficult to determine if the replica is suitable for a production restore. The following information can help you determine whether a replica is suitable for restore in its current state.

First of all, Replication Manager does not prevent the restore of a given replica, no matter what was done to it while it was mounted. That's because there could be reasons to perform a restore under almost any circumstances. Table 3 summarizes the cases where restore is *recommended* (safe restore situations), *suitable* (may require some manual adjustments on restore), or *not suitable* (the restore is not recommended in this situation). Each case includes a footnote providing more insight.

Table 3. Restoreability Impact

Footnote	Consistent split	Recovery type	DB rename ?	Restore advice
1	Yes or No	Filesytem only / Prepare Only	Yes or No	Recommended
2	Yes	Read Write	Yes	Not suitable
2	No	Read Write	Yes	Not suitable
3	Yes	Read Write	No	Not suitable
4	No	Read Write	No	Not suitable
5	Yes	Read Only	Yes	Not suitable
5	No	Read Only	Yes	Not suitable
6	No	Read Only	No	Recommended
7	Yes	Read Only	No	Suitable

1	No change is being made to the replica in that situation during mount, therefore a restore of such a replica is harmless and will restore the data from the replica, unaltered.
2	The database rename option alters the database in an invasive way. The database on the replica is no longer the same database. Restore is typically not suitable.
3	Even though the database rename option was not used, the database was opened in "Read/Write" mode after the recovery and therefore its data contents could have been changed. It is not suitable for restore unless the intent was to perform surgery on the database remotely and

	then restore the altered version to the production environment.
4	Similar to #3 with the added handicap that resetlogs was performed on this database, because of its non-consistent split, hot backup status. Because of that any archive logs from the production system will no longer be applicable and this limits the recoverability to just the point in time of the replica (including changes that have occurred on the mount host). Unless there is a specific circumstance justifying this, the restore is not recommended.
5	Even though the database was opened in "Read Only" mode, the db rename option has altered it in the same invasive way as the "Read/Write" mode does. The only difference is that the data contents of the database will not be changed. The change of the db name is enough to make it unsuitable for restore.
6	The database was recovered on the mount host but then opened in "Read Only" mode, preserving its data and incarnation status. This is a recommended candidate for a restore as it does not change the data but allows you to review the data. Note that for non-consistent split replicas, the backup control file is restored back in ERM_TEMP_BASE/‹sid›/. If the current control files are still available on the production system, EMC recommends their use. If they were lost, the backup control file may be copied in place and used for the recovery. Starting from Replication Manager version 5.4, the backup control file is copied to ERM_TEMP_BASE/‹db name›. **Note:** The backup control file contains references of the datafile paths changes as they were on the mount host and the user must manually update the paths (by using commands such as alter database rename file 'b' to 'a') during the restart of the production database. This replica is still suitable for restore.
7	Similar to #6. Consistent split replica contains the live current control files. If those are restored, they will be tainted with the datafile paths changed during the mount (control files reference datafile paths among other things). The user should manually update the paths (by using commands such as alter database rename file 'b' to 'a') during the restart of the production database. This replica is suitable for restore recoverability is not compromised.

Important: This table is also applicable to NFS replicas when the snap that was created was "Read/Write". For "Read Only" snaps, no action that is performed on the mount host affects the restorability of the snap.

Replication Manager replicas for repurposing

Mounting replicas with Replication Manager is traditionally used in two main scenarios:

- ***Backup***, where the end result is usually a backup to a tape or other media, attached to the backup server which is referred to as the mount host, from the Replication Manager perspective
- ***Repurposing***, where the replica is used as an entity of its own, for various purposes, such as reporting, testing or other types of data scrubbing which are either too I/O intensive or too risky to be done on the production system

Backup use cases are usually very straight forward in terms of Replication Manager's involvement during mount and do not require much knowledge of the mount options that have been described in the mount section of this document. Most of the times, the replica will be mounted without renaming the SID or database, with original path and no recovery (*do not perform database operations*, *or generate scripts for manual recovery* options).

Generally issues of multiple copies of data, path-collisions and other conflicts accompany the repurposing use cases. Many times, several replicas will be created from the same database, and mounted to a single mount host for a different "purpose".

The following table summarizes the types of precaution and Replication Manager mount options to consider when implementing repurposing use cases:

Replication Manager mount options and other considerations	Recommendations
Database name	Change the database name if you are going to have multiple copies of the same database concurrently running on the mount host. Oracle does not allow multiple copies of the same database name to run on the same host, even if they are in different locations.
Sid name	Change the sid (instance) name if you are going to have multiple copies of the same instance on the mount host. Oracle does not allow multiple copies of the instance to run on the same host. That is because the instance is

	the unique process realm by which a database is accessed. You need to change the sid name even if you are not planning to recover the database. That is because Replication Manager copies files that are SID-related into specific locations, such as $ORACLE_HOME/dbs/init‹sid›.ora or creates directories such as $ORACLE_HOME/admin/‹sid›/ etc. Keeping the SID unique will avoid collisions at several levels.
Alternate path	Alternate path options such as a path prefix, or the path mapping table, need to be used to allow multiple copies of the data to coexist on the mount host. These options are for filesystem-based replicas only. Without these options, mount points conflicts will occur way before any potential database recovery comes into play. Keep in mind that these alternate path options only work at the mount point level and are unaware of any sub directories or files that lay beyond the mount points. For example, the following path-mapping pattern is invalid: /oracle/data/sid/control01.ctl → /oracle2/data/sid2/control01.ctl (/oracle is the mount point). /oracle/data will be mapped to /oracle2/data but the sid directory will not be mapped to sid2 as it is not a mount point.
Rename ASM disk groups	This option is the ASM equivalent of the alternate path option, and only allows a prefix to be added to all diskgroups of the replica. Again, if multiple replicas need to coexist on the same mount host, this is a

	required option, to avoid path conflicts. Note that asmlib volume names (Linux) are taken care of automatically during every mount (by being renamed to randomly generated names). Replication Manager does not give access to any asmlib volume renaming option.
Customize Initialization parameters	This option is not automatically required when doing repurposing or multiple concurrent replicas. However, it is usually needed because of the fact that the resources on the mount host are not unlimited. In particular, parameters related to the sga size often need to be adjusted to accommodate several Oracle instances running at the same time. If the production host has 8gb of RAM and runs a 6g instance, and the mount host also has 8gb of RAM, it will run one replica without any problem. However, a second instance would exhaust resources on the host. Therefore in this example, if the target needs to run 2 copies of database on the mount host, it would be suitable to set the sga size to 3gb each instead of 6.
Fail mount if SID exists	This option acts as a safety check to avoid overwriting SID-specific files on the target host. It is recommended to leave that option set to true, especially when dealing with multiple copies of the same data, where a mistake of supplying the same SID twice can easily happen. In this case, the second replica being mounted with a SID that already exists on the host would fail instead of overwriting initialization files and password files

	and other instance related components.

Note about TDE: If Transparent Data Encryption is enabled on the production database, the security administrator is responsible for importing and opening Oracle wallets on the mount host if access to encrypted data is required in the repurposing use case.

Replication Manager and Oracle Transparent Data Encryption (TDE)

Oracle TDE technology is an encryption capability, at the column, table or tablespace level, managed by a decryption key traditionally stored in a password protected wallet. Replication Manager can perform its basic use cases of replication, mount (including recovery of the database) and restore of databases that have TDE enabled.

Replication Manager does not integrate in with TDE and does not open, close or manipulate Oracle wallets in any way. Consequently it does not participate in accessing data inside the tablespaces that are replicated. If access to the encrypted data is required on the mount host, the security administrator is responsible for importing and opening the Oracle wallet prior to accessing the data.

Application set and job simulations

Replication Manager users can ascertain if there are any configuration issues associated with any job by running a simulation of the job before running the job that actually creates the replica. For the most part, this functionality helps users quickly determine the health of the configuration, without having to wait for what is usually the longest part of the replication, the synching of the source devices with their target counterparts.

Replication Manager performs four main categories of actions during a replication, mount, or restore:

- Queries to the database or to the array range from queries to the Oracle database, such as “select tablespace_name, status from dba_tablespaces” to symrslv calls to translate devices into LUNs and so on.
- File creation includes creating backup control files and copying archive logs.
- Mirror operations include splits, fractures, and synchs, establish, reverse sync, and other array calls that affect the status of the source and target devices.
- Cataloging includes updates to the catalog and metadata information at the end of the replication, as well as sending files described in bullet 2 over the network to be part of the replica.

For Oracle simulations, the first bullet is completed and the other three are not performed. In other words, anything that is actually altering the state of the system is not performed during a simulation. Rather, the entire flow of the job is run and all the

checks, queries, software version verification, and credential validation are completed.

Because the last three items are skipped during a simulation, there are certain errors that may still occur during the actual run of the job, which did not happen during the simulation. These include:

- Problems that occur while selecting or establishing mirrors
- ERM_TEMP_BASE not having enough free space to contain the new generated files
- Problems caused by full archive log directories

Conversely, the following problems would be detected at simulation time:

- General connectivity problems to the database.
- Listener issues
- Access issues (such as user or password mismatches)
- Offline tablespace or datafiles
- Required components not located on a file system or device that resides on the array.
- Various problems related to array connectivity

Troubleshooting Oracle issues during replication

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

For automatic discovery of the Oracle databases several Oracle queries are run at the start of the replication. Whenever Replication Manager operations fail with Oracle SQL errors it is important to get the result of some of the queries.

Connection failure

When the Replication Manager agent can't connect to the Oracle database at configuration or replication time, check the following aspects of your environment:

1. Determine if the TNS listener or Oracle Name Service is up and running.
2. Run `tnsping <service identifier>` to make sure that Oracle connection works.
3. Make sure the Oracle service is up and running. Run the SQLPLUS command:
   `select * from dba_data_files`

Unable to create the backup control file

If Replication Manager is unable to create the backup control file, follow these steps to troubleshoot the issue:

1. Make sure that the /tmp directory is not full.
2. Set the environment variables ORACLE_SID and ORACLE_HOME.
3. Run the command $ORACLE_HOME/bin/sqlplus ‹user/password @ Oracle Service› as sysdba.
4. Alter the database backup control file to $ERM_TEMP_BASE/filename.

   (where ERM_TEMP_BASE could be /tmp or any other location specified in rc.irclient)
5. In RAC configurations only, this error can occur when the application set was defined against the database rather than a specific SID corresponding to one of the RAC nodes.

Some important Oracle errors and possible causes

Table 4 shows some important Oracle errors and provides possible causes and solutions to these error situations.

Table 4. Oracle errors, causes, and solutions

Error	Cause/Solution
ORA-12154: TNS could not resolve the service name	Identify where the TNS configuration files can be found on the host. Check that the TNSNAMES.ora file is in the default location or if not, update the TNS_ADMIN field accordingly on the "Create Application Set" screen. Check the SQLNET.ora file and identify the order for the service resolution is as follows: TNS, ONAMES Check the TNSNAMES.ora file and see if the service is correctly defined. Run tnsping for the service and check the output for the hostname and port number.
ORA-12560: TNS protocol adapter error	Check whether tnsping for the service identifier works correctly.

Error	Cause/Solution
ORA-12505: "TNS:listener does not currently know of SID given in connect descriptor"	This means that the listener that handled the connection request doesn't know of the SID described in the descriptor defined in the tns alias in tnsnames.ora. This can happen especially in "offline" situations such as offline replications or restore, where Replication Manager has to connect back to an instance that is in idle state. Make sure that the listener being started is the expected one. By default, Oracle uses a listener called "LISTENER". So commands such as lsnrctl start/stop with no argument will act on the default listener. If the listener defined in listener.ora is not referred to by the default name (for example, LISTENER_RAC), then its explicit name needs to be mentioned in the listener commands (lsnrctl start LISTENER_RAC, lsnrctl status LISTENER_RAC). The way to verify that the listener is listening to the SID requests is by running the status command: lsnrctl status ‹listener name› or lsnrctl services ‹listener name›.

Conclusion

This white paper explains how to use Replication Manager with Oracle Database Server, including how to configure your Oracle environment for optimal replica creation. The paper also describes the details associated with creating an application set and a job to replicate Oracle data with Replication Manager. These sections describe the details of how to create a replica, including important information about setting credentials, discovering the Oracle environment, planning your replica layout, and so on. In addition, the white paper reviews consistency settings and optional files that Replication Manager can replicate.

Other parts of the paper describe such advanced Oracle subjects as replicating the Flash Recovery Area, replicating data stored under an Automatic Storage Management instance, and restoring Oracle data in varied Oracle configurations.

The white paper describes various mount techniques in detail, including specifics of how Replication Manager carries out mounts such as a mount to the production host. The mount section also describes how various mount options affect the resulting mount and there is also a section devoted to the impact of mount on future restores of the mounted replica. Finally, a troubleshooting section provides solutions to some issues that Oracle users might encounter when replicating their data with Replication Manager.

References

For more information on Replication Manager and how to use it in your Oracle environment, consider the following sources:

- EMC Replication Manager Product Guide
- EMC Replication Manager Administrator's Guide
- EMC Replication Manager Release Notes

For more information on replication technologies in conjunction with Oracle Block Change Tracking, consider the following source:

- Reducing Backup Window and Recovery Time with Oracle Database 11g RMAN and EMC TimeFinder/Clone